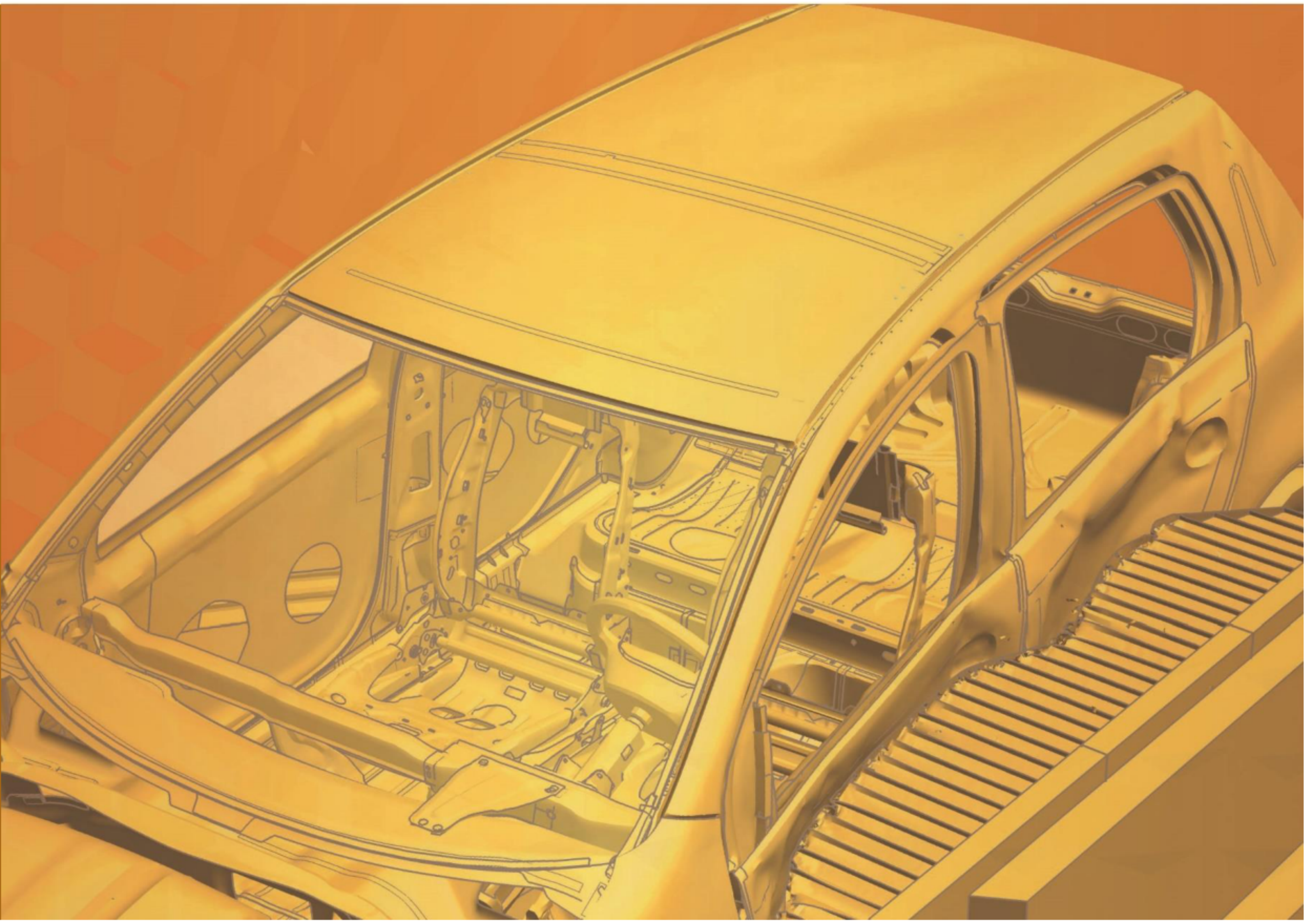
D3PLOT

D3PLOT Manual

from Oasys Ltd



Version 21.1

For help and support from Oasys Ltd please contact:

UK
Tel: +44 121 213 3399
Email: dyna.support@arup.com
China
Tel: +86 21 3118 8875
Email: china.support@arup.com
India
Tel: +91 40 69019723 / 98
Email: india.support@arup.com
USA West
Tel: +1 415 940 0959
Email: us.support@arup.com

Web: www.arup.com/dyna

or contact your local Oasys Ltd distributor.

D3PLOT 21.1

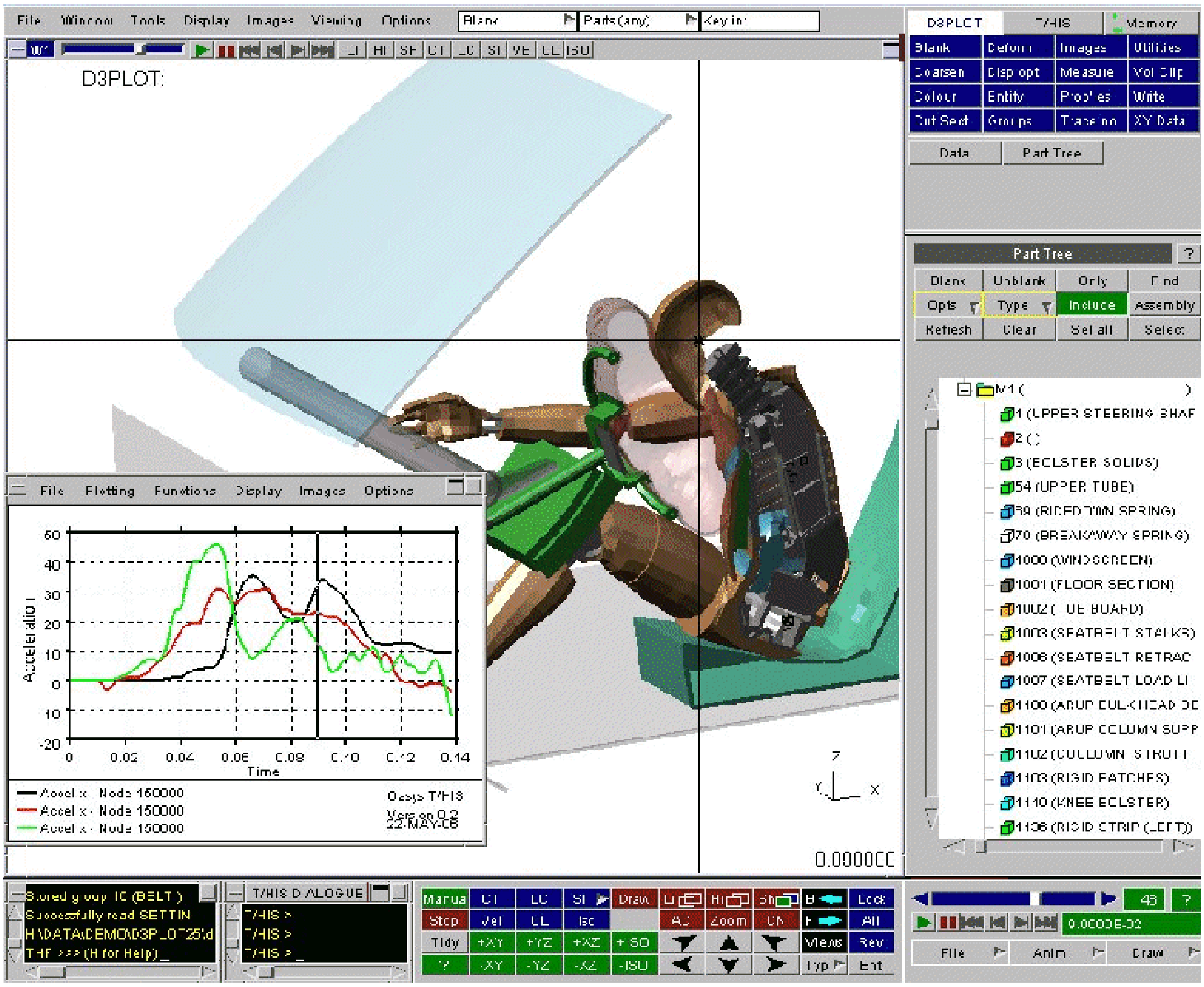
Table of Contents

D3PLOT

1. D3PLOT

D3PLOT Software Manual
from Oasys Ltd

D3PLOT



POST-PROCESSING OF MODELS WITH SPECIALIST FUNCTIONS

Oasys Limited | 8 Fitzroy Street | London | W1T 4BJ

2. Preamble

2.1. Acknowledgements

Acknowledgements

The names **LS-DYNA** , **LS-PREPOST** and **LS-OPT** are all registered trademarks of ANSYS, Inc. and are used in this manual by permission.

2.2. Abstract

Abstract

Transient analyses make much more sense when one is able to see how the results change with time. Most existing post-processors only allow you to draw one image at a time and, while it is possible to assemble a sequence, producing a set of results can be very tedious.

This code allows you to access the LS-DYNA database directly and to draw line, hidden-line, continuous-tone, line contour, velocity arrow, greyscale and shaded-image plots for any results state in the file. It also allows you to store these images in the display device memory and to redraw them in sequence and so to produce animated graphics.

2.3. Host Computers

Host computers

The code is available for all commonly used operating systems: Windows, Unix and Linux in 32 and 64 bit modes. It is available on all common work-stations and mainframes.

2.4. Memory Requirements

Memory requirements

Memory is allocated dynamically, so the amount required rises in proportion to the problem size. However machines with less than 64 MBytes of physical memory (RAM) are unlikely to function satisfactorily.

2.5. Output Devices

Output devices

The code supports the following graphics devices:

OpenGL	3-D, hardware assisted graphics
Stereo	Stereo OpenGL on hardware that supports this

Images may be captured in the following formats:

Animated "movie" formats:

AVI	AVI animation files
GIF	Animated GIF files
MP4	MP4 animation files

Static "image" formats

BMP	BMP (bitmap) static image files
JPEG	JPEG static image files
PNG	PNG (Portable Network Graphics) static image files
GIF	GIF (Graphics Interchange Format) static image files
Postscript	Colour and greyscale laser plotting
PDF	Colour and greyscale Portable Document Format (PDF) output

External animations and static images may also be imported for display in the following formats:

BMP
JPEG
PNG
GIF

}
} For display of static images
}

AVI For display of animated images

2.6. Text Conventions Used in this Manual

Text conventions used in this manual

TYPEFACES:	Three different typefaces are used in this manual:
Manual text	This typeface is used for text in this manual.
Computer type	This one is used to show what the computer types. It is also used for equations etc.
Operator type	This one is used to show what you must type.
Button text	This one is used for screen menu buttons (eg **APPLY**)

NOTATION:	Triangular, round and square brackets have been used as follows:
Triangular	To show generic items, and special keys. For example: **<list of integers> <filename> <data component>** **<return> <control Z> <escape>**
Round	To show optional items during input, for example: < **command** > (<optional command>) (<optional number>) And also to show defaults when the computer prompts you, eg: **Give new value (10) :** **Give data component (FX_AXIAL_FORCE) :**
Square	To show advisory information at computer prompts, eg

```
Give terminal type [M for list] :
```

```
D3PLOT_MANAGER >>> [H for Help] :
```

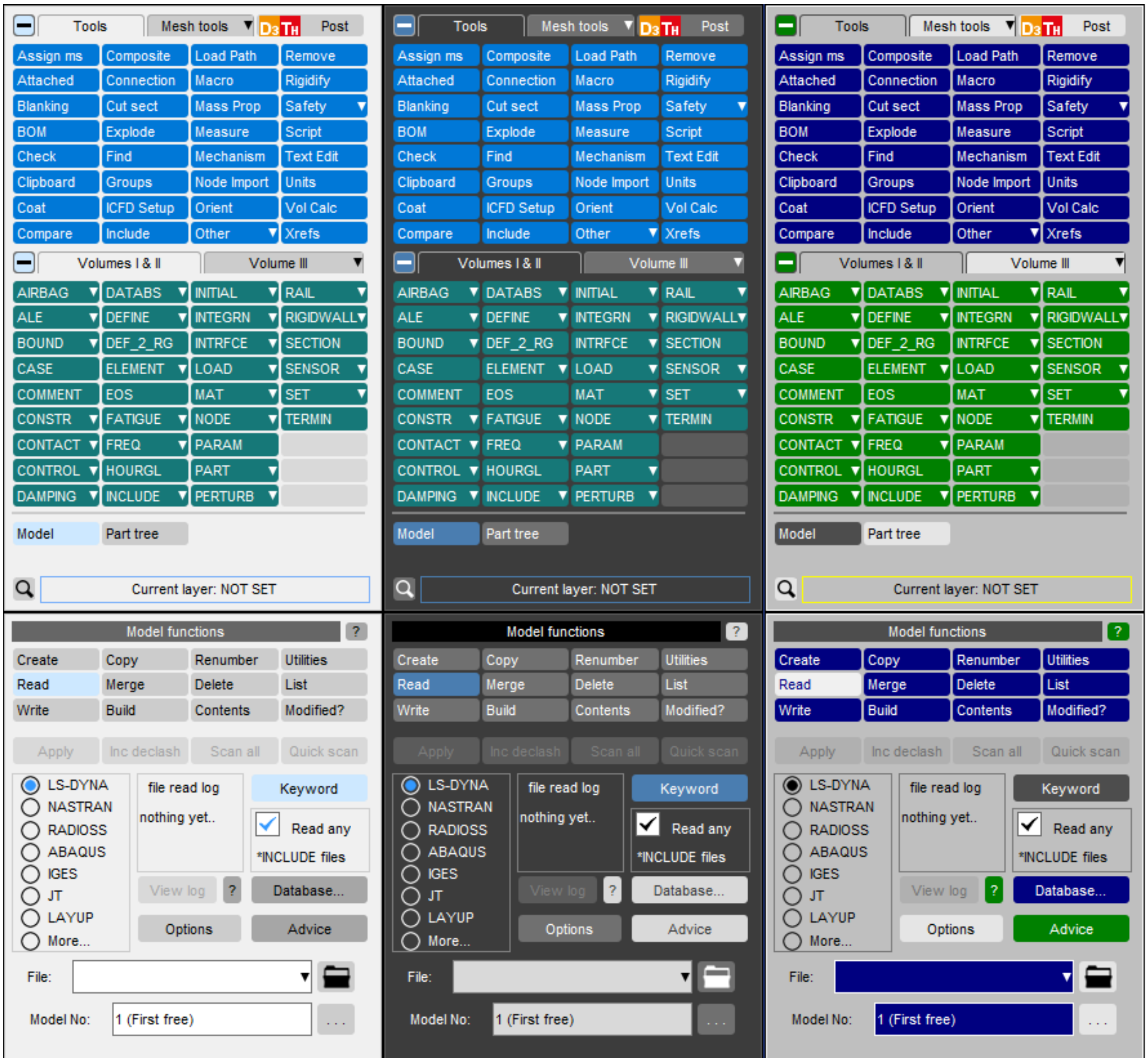
Also to show implicit commands, eg

```
[WRITE] SCAN <entity> <number of values>
```

3. Themes for the Graphical User Interface

Themes for the Graphical User Interface

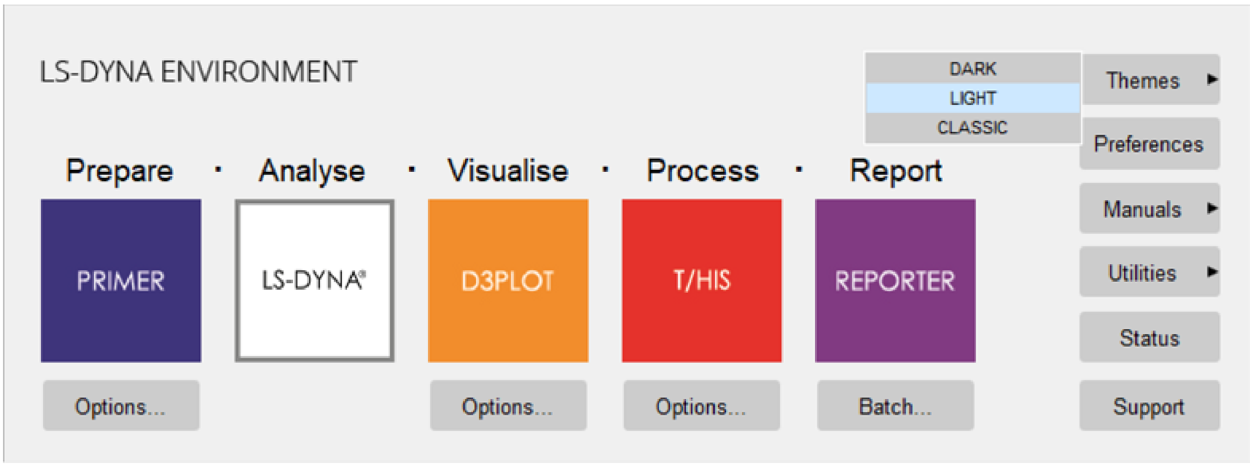
In addition to our Classic GUI theme, beginning in Oasys Suite 17.0, users can select either a Light or Dark theme. Both of these provide a more modern look and feel for the software, as well as offering different colour and contrast options for comfort and accessibility.



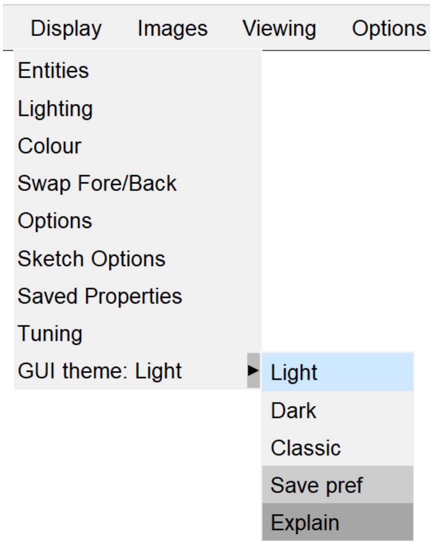
3.1. Setting the Theme

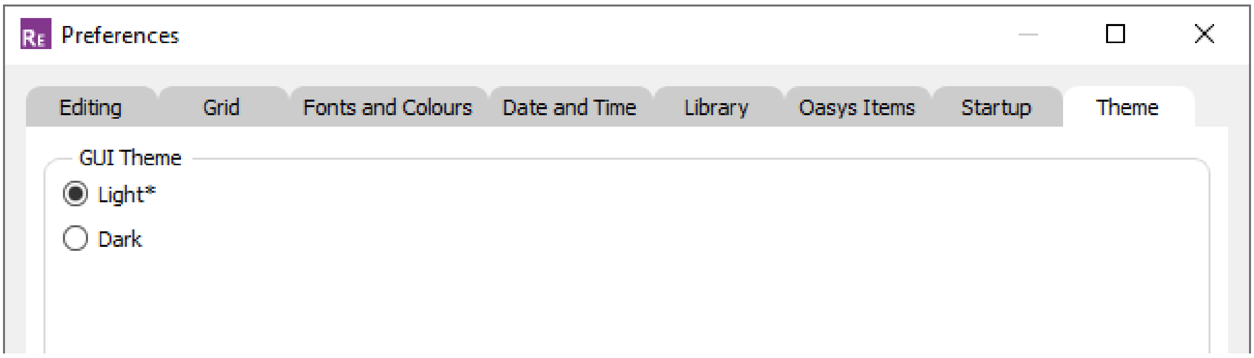
Setting the theme

The default software theme in Oasys Suite 21.1 is Light. This can be changed from the Oasys SHELL by choosing from the **Themes** pop-up. This automatically saves the selected theme as your preference for all programs.



The theme can also be set for individual programs from the **Display** menu in PRIMER, D3PLOT and T/HIS or the **Preferences** menu (**File**->**Preferences...**) in REPORTER. This choice is not automatically retained after exiting the program, so you must select a theme, then select **Save pref** to ensure a theme is used for all future sessions.




Preferences
Editing
Grid
Fonts and Colours
Date and Time
Library
Oasys Items
Startup
Theme
GUI Theme
Light*
Dark

4. Supported LS-DYNA Features

4.1. Element Types

Element Types

The following entity types in LS-DYNA can be processed in D3PLOT:

Solid Elements	8 noded "bricks" (hexahedra), and the elements they degenerate to: "wedges" and tetrahedra.
Thin Shells	4 noded quadrilaterals and 3 noded triangles.
Thick Shells	8 noded shells.
Beams	2 noded beams.
Discrete elements	Springs, dampers, lumped masses.
Seat-belt elements	Belts themselves, slip-rings and retractors.
SPH elements	Smooth Particle Hydrodynamic "sphere" elements (V9.3 onwards)
Airbag Particles	The particles used in the Airbag Particle inflator method (V9.3 onwards)
DES elements	Discrete Element Spheres (D3PLOT 12.0 onwards)
Contact surfaces	LS-DYNA models impact, friction, sliding, etc by the use of contact surfaces that can be thought of as two-dimensional elements overlaying the surfaces of solid and/or shell models. These are not true elements, but rather sub-areas, so to prevent confusion they will be referred to as interface "segments" from now on.
Spotwelds	*CONSTRAINED weld types, and also spotweld beams, solids and solid hex clusters
SPCs	The SPCs themselves and their reaction forces
Using the .XTF file from LS-DYNA	In LS-DYNA joint, lumped-mass and stonewall geometries are sent to the `.XTF` file and so may be recovered for plotting in D3PLOT. However results from these are not available for plotting in D3PLOT: they may be viewed in XY plot form in T/HIS.
Alternatives to the .XTF file when using MPP LS-DYNA	MPP LS-DYNA, and also SMP versions from ls970 onwards can also generate a "binout" (or LSDA) file; and the MPP version cannot generate a `.XTF` file. D3PLOT does not read this file

directly, but from V90 onwards the information previously extracted from the `.XTF` file is now available from the `.ZTF file` - see below.

Using the .ZTF file

PRIMER can generate a pseudo-database .ZTF file directly from the input deck. This is intended to contain extra information not in the normal LS-DYNA database files, and also to replace the .ZTF file.

In D3PLOT 8.3 the `ZTF` file allows you to visualise:

- Nodes on "nodes_to_..." contacts.
- Nodal constraints and restraints
- Spotweld beams.

From D3PLOT 9.0 onwards you may also visualise the following even when the .XTF file is missing:

- Stonewalls (rigid walls)
- Springs and dampers
- Seatbelt elements, retractors, slip-rings and pre-tensioners
- Joints
- Lumped masses

In addition the names of parts and contacts, previously stored in the .XTF file, are also available.

From D3PLOT 9.2 onwards you may also visualise:

- Beam "true" sections
- Part tree: organised by include files, assemblies and sub-assemblies (as in PRIMER)

From D3PLOT 9.3 onwards:

- Discrete and Seatbelt elements can be processed by PART
- Part, Part_composite and Section data are available

In addition ls-dyna cut-section definitions (*DATABASE_CROSS_SECTION) may be used to defined D3PLOT cut sections

From D3PLOT 9.4 onwards:

- Spotwelds (Constrained, beam, solid, solid cluster)
- SPCs

  The results for spotwelds, discrete elements, seatbelt and SPCs may also be displayed if a "binout" (LSDA) file is present.

From D3PLOT 10.0 onwards you may also visualise:

- PRIMER Rigid and Deformable connections using a new "Bolt" symbol.
- MIG lines created in PRIMER.
- *DATABASE_CROSS_SECTION definitions and locations.

  The results for *DATABASE_CROSS_SECTION may also be displayed if a "binout" (LSDA) file is present.

From D3PLOT 16.0 onwards you may also visualise:

- *LOAD_NODE_POINT, *LOAD_NODE_SET, *LOAD_BEAM, *LOAD_BEAM_SET, *LOAD_SHELL, *LOAD_SHELL_SET, *LOAD_SEGMENT and *LOAD_SEGMENT_SET definitions. Visualisation is currently restricted to the direction and location of loads and does not include the magnitude of the loads.
- Composite data for shells, including layer and ply information if present
- Shells and thick shells will also "know" how many integration points they have through their thickness

If the T/HIS link is invoked then elements and nodes in time-history blocks can be displayed, and screen picked for time-history plotting.

Data for other entities are not sent to any database files, so they are not displayed.

D3PLOT is primarily for post-processing results from LS-DYNA, but results from TOPAZ3D (thermal analysis) and NIKE3D (implicit structural analysis) may also be processed. Both codes write a subset of the entity types listed above.

4.2. Types of Results Processed by D3PLOT

Types of results processed by D3PLOT

LS-DYNA is a three-dimensional non-linear analysis code which models the transient behaviour of structures in the time domain. The output you may process graphically in D3PLOT is shown by category below, together with the intrinsic coordinate system in which it is written by LS-DYNA.

ENTITY TYPE	DATA GENERATED	COORDINATE SYSTEM
Nodes:	Coordinates Velocities Accelerations Temperatures	Global cartesian " " " " [none]
Solids: (1)	Stress tensor Plastic strain Strain tensor (optional)	Global cartesian [none] Global cartesian
Thin shells: (2)	Stress tensor Strain tensor (optional) Forces and moments Plastic strain Thickness Strain energy density	Global cartesian " " Element local [none] [none] [none]
Thick shells: (2)	Stress tensor Plastic strain Strain tensor (optional)	Global cartesian [none] Global cartesian
Beams: (3)	Forces and moments Plastic data (optional) Stress/strain data (optional)	Element local " " " "
SPH elements: (4)	Stress tensor Plastic strain Also:	Global cartesian [none]

	Strain tensor Density Pressure Internal energy Radius of influence	#neighbours	Global cartesian [none] " " " " " "
Airbag particles: (4)	Mass Radius Spin energy Trans energy Distance to nearest segment Coordinates Velocities	Also: Gas ID Leakage state	[none] " " " " " " " " Global cartesian Global cartesian
Spotwelds: (5)	Axial force Shear force Failure status Failure time		Element local " " [none] " "
Springs and dampers: (5)	Force Elongation Moment Rotation		Element local " " " " " "
Seatbelt elements: (5)	Belt force Belt length Slipring pull-through Retractor force Retractor pull-out		Element local " " " " " " " "
SPC forces: (5)	Forces and Moments		Global Cartesian
Cross sections (*DATABASE_CROSS_SECTION): (5)	Forces and Moments		Global Cartesian
Contact surfaces	Contact stresses Contact forces		Segment local Global cartesian

(1) Results for solids are by default written by LS-DYNA at the element centre only, even if an element formulation with > 1 integration point is used. Solid results can be written at all integration points for element formulations with > 1 integration point by setting NINTSLD=8 on the *DATABASE_EXTENT_BINARY card. Solid results from NIKE3D are written at all 8 integration points.

(2) By default shell stress and strain tensor results are written at top and bottom integration points, and stresses also at the neutral axis. Data output at more than these 3 points through the element thickness may be selected, and will

be available for display if present. Fully integrated shells in ls-dyna with more than 1 integration point on plan still only write (averaged) data at the element centre.

(3) As well as basic forces and moments extra "plastic" data from resultant beams, and data at integration points for integrated beams are supported.

(4) SPH and Airbag particle data are only processed from D3PLOT 9.3 onwards.

(5) These elements and data components are only processed from D3PLOT 9.4 onwards, and they require a ZTF file to provide geometry (for display) and a binout (LSDA) file if results are to be extracted.

D3PLOT will generate derived data components, (eg von Mises, principal, etc) from the above, and will also transform results from global to local coordinate systems if required.

In addition the following written output may be generated for more "global" model data.

ENTITY TYPE	DATA COMPONENT	COORDINATE SYSTEM
Whole model	Average velocity & momentum Kinetic and internal energies Mass	Global cartesian [none] [none]
Each material	Average velocity & momentum Kinetic and internal energies Mass	Global cartesian [none] [none]
Contact surfaces	Summary forces	Global cartesian
Airbags (of Airbag particles)	Volume	[none]
Stonewalls	Normal force	Local vector

Any data component written for nodes or elements that is actually present in the database files may be plotted graphically, presented as an X-Y plot of <data> vs <time> (or vs <data>), and written out in tabular form. This is also true of components derived from the basic ones.

In addition many geometric and topological attributes of nodes and elements (eg material number, elements connected to nodes) may be tabulated.

Any scalar data component may be "scanned" for maximum / minimum values, and tables of the top and bottom values produced.

Although their topology is extracted and they are displayed visually the results for springs, joints, seat-belts etc, are not available for plotting in D3PLOT. This is because these data are not currently available in the appropriate database files.

4.3. Support For LS-DYNA Multiphysics Solvers

Support for LS-DYNA Multiphysics Solvers

In additional to the standard structural solver recent versions of LS-DYNA now have additional Multphysics solvers. D3PLOT 12.0 supports these additional solvers and can read and display results from the following solver types.

- Incompressible CFD (ICFD)
- Compressible CFD (CESE)
- Electromagnetic (EMAG)

In order to display results from these solvers an additional file called "multiphysics.components" must be present in the directory containing the D3PLOT executable.

In D3PLOT 12.0, results from all 3 solvers can be plotted using any of the standard plotting modes (CT, SI, LC, ISO, CL, VEC - See Drawing commands that plot data for more details) but support in other menus is limited. At present ICFD, CESE and EMAG results are not available in either the WRITE (see WRITE Listing numerical data to screen and/or file) or XY-DATA (see XY_DATA Drawing numerical data as XY plots and/or writing it to file) menus.

For more information of the data components available for the multiphysics solvers see Data components for Multiphysics solvers

4.3.1. Multiphysics Parts

Multiphysics Parts

Multiphysics results are grouped together in the PTF file into what are known as domains. Each analysis can contain multiple domains which contain either surface (boundary) data or volume data. Each domain can also contain multiple parts.

As it is possible for an analysis to contain both structural parts and multiphysics parts with the same ID D3PLOT separates the multiphysics parts from the structural parts and then further sub-divides the multiphysics parts into ICFD, CESE and EMAG and then into surface and volume parts.

e.g	P3	Part 3 (structural)
	IC_S_P3	ICFD Surface Part 3

IC_V_P10	ICFD Volume Part 10
CE_V_P10	CESE Volume Part 10
EM_S_P10	EMAG Surface Part 10

At present complete ICFD, CESE and EMAG parts can be blanked/unblanked but is not possible to blank individual ICFD, CESE or EMAG elements.

4.3.2. Multiphysics Nodes and Elements

Multiphysics Nodes and Elements

As with part ID's the elements and nodes within domains can share ID's with structural nodes and elements. To distinguish between the nodes and elements belonging to the multiphysics domains they are labelled using the prefix Dn where "n" is the domain number.

e.g	N900	Node 900 (structural)
	D2/N900	Node 900 in domain 2

At present multiphysics nodes can be used to locate cut sections but they can not be selected in other menus.

4.4. D3PLOT Representation of Elements and Other Entities

D3PLOT Representation of Elements and Other Entities

The three figures below show examples of how each of these types appear as drawn by D3PLOT. They also show the labelling conventions used:

Entity Type	Labelled

Solids
Thin Shells
Thick Shells
	`H...`
	`S...`
	`T...`

Beams
Springs / Dampers
Seat-belts
Slip-rings
Retractors
Lumped-masses
	`B...`
	`SP..`
	`SB..`
	`SR..`
	`RT..`
	`LM..`

SPH Elements Airbag Particles	These elements may be drawn as points, cubes or spheres. This is controlled in Display Options	**HP..** **AP..**
Spotwelds of various types:	*Constrained_Weld *Constrained_Generalized_Weld *Element_Beam (spotweld beam) *Element_Solid (isolated solid spotweld) *Define_Hex_Spotweld_Assembly (solid spotweld cluster)	**CW..** **GW..** **BW..** **HW..** **HA..**
PRIMER Connections	Rigid Bolts Deformable Bolts	**BR..** **BB..**
SPCs		**SPC..**
Contact Segments Stonewalls Joints		**I...** **W...** **J...**
Nodes		**N...**
Cross Sections (*DATABASE_CROSS_SECTIONS)		**XSEC...**
Loads of various types:	*LOAD_NODE_POINT *LOAD_NODE_SET *LOAD_BEAM *LOAD_BEAM_SET *LOAD_SHELL *LOAD_SHELL_SET *LOAD_SEGMENT *LOAD_SEGMENT_SET	**LND..** **LNDS..** **LBM..** **LBMS..** **LSH..** **LSHS..** **LSE..** **LSES..**

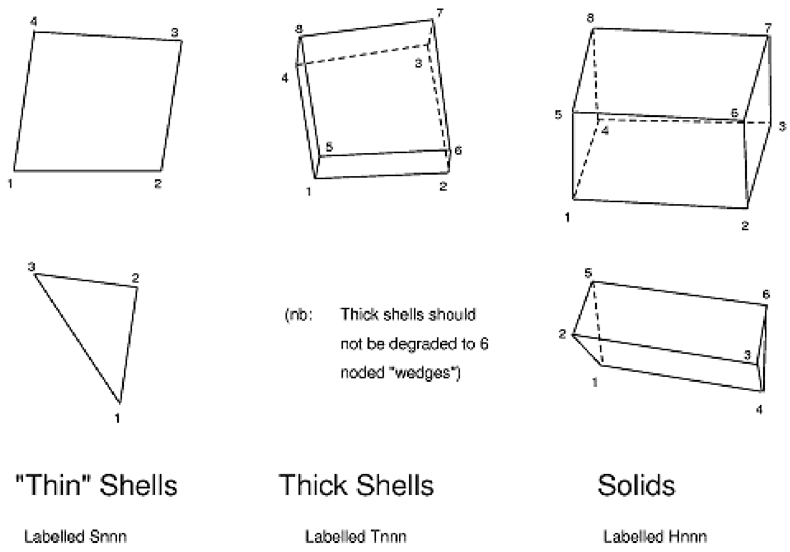
Note the following:

a) Arbitrary numbering of nodes, elements and materials in LS-DYNA is supported. This covers nodes, solids, shells, beams, springs, seat-belt types and lumped-masses. Joints, stonewalls and contact segments are all numbered sequentially from 1.

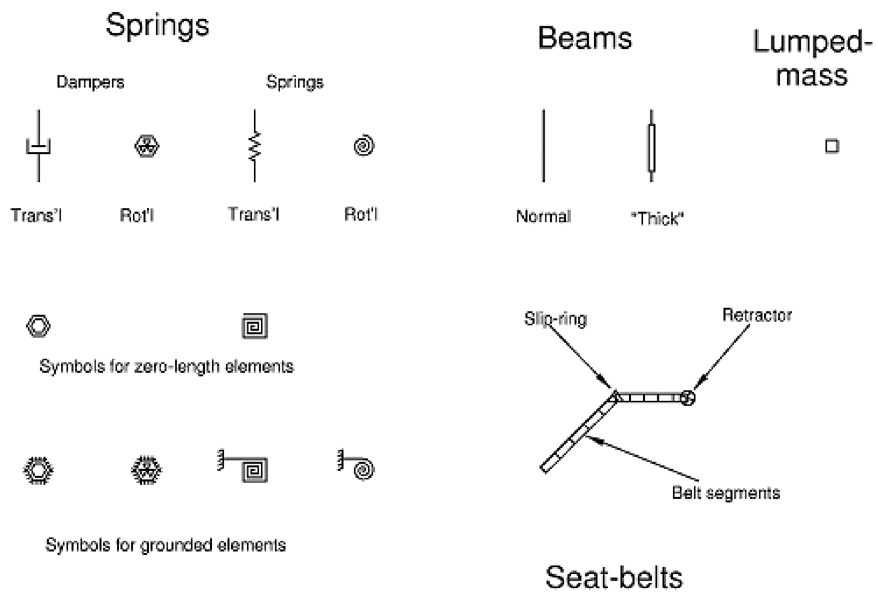
b) Springs, seat-belt types, lumped-masses, joints and stonewalls are only recovered and drawn if an "extra time-history" (**.XTF**) file is found - this file is optional. Only the topologies of these elements are extracted: use T/HIS to extract and plot time-history results for these elements.

c) Contact segments are only recovered and drawn if a "contact force" (**.CTF**) file is found - this file is optional.

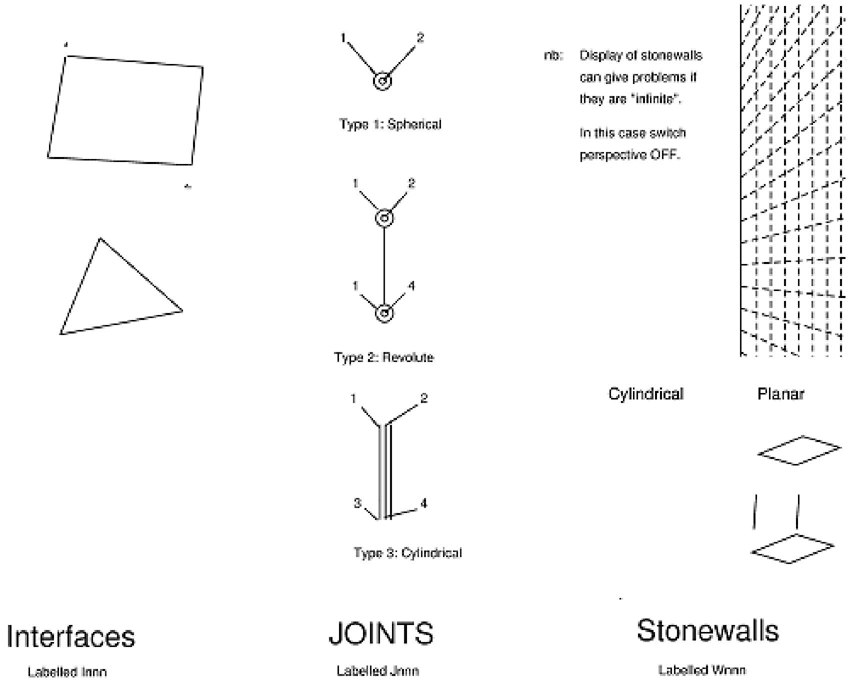
d) These figures show the symbols used on 2D devices. When 3D graphics is used some symbols are slightly different: springs become a spiral, damper symbols become a three-dimensional dashpot, joint circles become spherical, "thick" beams have rectangular sections. This is done to make symbols meaningful regardless of how the view is oriented in 3D space.



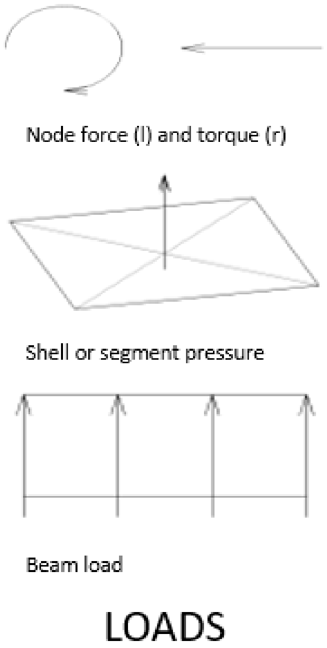
Solids shells and thick shells



Springs, beams, lumped masses and seatbelt elements



Contacts, stonewalls and joints



Loads

4.5. LS-DYNA Output Files Processed

LS-DYNA output files processed

D3PLOT reads the LS-DYNA binary database files directly. No intermediate translation is required and results can be viewed while an analysis is running.

To open the files see Basic Data Extraction and Plotting .

The database files processed are (see Contents of the LS-DYNA database files processed by D3PLOT for more details):

4.5.1. Complete State (Plot) File

Complete state (plot) file

```
<name>.ptf or d3plot
```

This file contains the undeformed geometry of the model, followed by complete dumps of its current geometry. It contains information about:

Nodes:	Coordinates, velocities, accelerations, temperatures.
Solids:	Stresses, strains, (extra data).
Beams:	Forces, moments, plastic rotations, plastic strains.
Thin shells:	Stresses, strains, force & moment resultants, strain energy density, thickness, (extra data).
Thick shells:	Stresses, strains, (extra data).
"Global" data:	Material energies, masses and velocities; normal force on stonewalls.

The file also contains information about deleted elements if the relevant material models and/or contact surfaces are used.

4.5.2. Dynamic Relaxation File

Dynamic relaxation file

```
<name>.rlf or d3drlf
```

This file contains the same information as the complete state file (.ptf) descibed above, but pertains to a dynamic relaxation analysis.

4.5.3. *FREQUENCY_DOMAIN Files

*FREQUENCY_DOMAIN files

Frequency domain analyses can be carried out in LS-DYNA to output the following files which D3PLOT can read. Some versions of LS-DYNA sometimes write data contigously across file boundaries when they should not, which prevents D3PLOT reading the results. Such results may be read by setting the environment variable PTF_CONTIGUOUS to TRUE (see APPENDIX D).

`<name>.d3eigv` or `d3eigv`	Modal results from a *FREQUENCY_DOMAIN analysis.
`<name>.d3ssd` or `d3ssd`	Results from a Steady State Dynamics analysis (*FREQUENCY_DOMAIN_SSD). If the <BINARY> flag on the *DATABASE_FREQUENCY_BINARY_SSD card is set to 2, then the file will also contain phase angle data.
`<name>.d3psd` or `d3psd`	Power Spectral Density results from a Random Vibration analysis (*FREQUENCY_DOMAIN_RANDOM_VIBRATION).
`<name>.d3rms` or `d3rms`	Root Mean Square results from a Random Vibration analysis (*FREQUENCY_DOMAIN_RANDOM_VIBRATION).
`<name>.d3ftg` or `d3ftg`	Results from a Random Vibration Fatigue analysis (*FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE).
`<name>.d3spcm` or `d3spcm`	Results from a Response Spectrum analysis (*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM).
`<name>.d3acs` or `d3acs`	Results from a frequency domain finite element acoustic analysis (*FREQUENCY_DOMAIN_ACOUSTIC_FEM).

4.5.4. Extra Time-History File

Extra time-history file

`<name>.xtf` (See also under T/HIS link below)

This file contains miscellaneous "time-history" data about the model. D3PLOT reads only the following basic topology and coordinates from it:

Springs:	Spring, damper & seat-belt geometry.
Seat-belt types:	Seat-belt, retractor and slip-ring geometry.
Lumped-masses:	Geometry and mass.
Joints:	Geometry and type.
Stonewalls:	Geometry, mass and topology.

This file also contains the names of parts and contacts, which will be displayed in menus if available.

4.5.5. Interface Force Files

Interface force files

`<name>.ctf or ctfile`	Interface force file
`<name>.blstfor or blstfor`	Blast force file
`<name>.fff or fsifor`	Fluid-Structure Interaction force file
`<name>.cpm or cpmfor`	Corpuscular Particle Method force file
`<name>.dem or demfor`	Discrete Element Method force file

These files contain information about contact surfaces:

Contact facets:	Topology, contact stress.
Nodes on facets:	Contact forces.

4.5.6. Extra "Static" Database File

Extra static database file

`<name>.ztf`

This file is not generated by LS-DYNA itself, but rather by running PRIMER on the relevant input deck. (This can be done automatically from the Shell).

It is a "static" file (ie no time-history data) that contains information about:

Nodes on NODE_TO_... contacts	The nodes on the "nodal" side of these contacts will become visible as "diamond" symbols. However no force data is recovered on these nodes.
Nodal restraints & constraints	For each of the six degrees of freedom of every node any restraint or constraint due to SPCs, *CONSTRAINED items, rigid bodies, inclusion on TIED or CONSTRAINT contacts, etc is stored. These DoFs can be displayed.
Beam section data	By storing beam element section data it becomes possible for D3PLOT to "know" which beams are spotwelds (ie section type 9 using *MAT_SPOTWELD) and to draw them as spotwelds.
Part and Section data	From D3PLOT 9.3 onwards, the *PART(_xxx) and *SECTION_xxx cards are written verbatim to the .ZTF file, making it possible to extract thickness and layer information. In addition it becomes possible to associate the Part ids of Springs and Seatbelt elements with those used by solids, shells and beams making it possible to operate on these "by part".
Load data	From D3PLOT 16.0 onwards, the *LOAD_NODE_POINT, *LOAD_NODE_SET, *LOAD_BEAM, *LOAD_BEAM_SET, *LOAD_SHELL, *LOAD_SHELL_SET, *LOAD_SEGMENT and *LOAD_SEGMENT_SET data are written to the .ZTF file, so that loads can be displayed on the model.

The ZTF file also contains information about section and material cards, coordinate systems, shell integration points, composite and beta angle data which does not translate directly to things which are drawn but nevertheless make it possible to display information "intelligently". In particular when used in conjunction with the binout file below it opens up the possibility of post-processing both visually and numerically a wealth of information that otherwise would not be accessible.

From D3PLOT 9.0 onwards, where the .XTF may not be present, this file also contains all the information previously stored in the .XTF file so that there is no loss of functionality.

4.5.7. BINOUT (LSDA) File

BINOUT (LSDA) file

Usually just " `binout` "

This is a file generated by LS-DYNA which contains the "time-history" information normally written to ASCII database files via the *DATABASE_ABSTAT etc keywords, but in binary form. It is created if the < **`binary`** > flag on the relevant card is set.

Only data of the specified types (eg ABSTAT for airbag data) is output.

D3PLOT 9.4 onwards reads this file and extracts the subset of its data that it can process, at present limited to:

- SPC results
- Spring and damper forces and moments
- Seatbelt and related element results
- Spotweld results
- *DATABASE_CROSS_SECTION results

Since the binout file contains only results, and not geometry information, D3PLOT also requires a ZTF file from PRIMER to specify the topology and geometry of these items, and without a ZTF file the binout file cannot be processed.

Binout files can be extremely large, and scanning their contents can take a significant amount of time - sometimes several minutes. In order not to slow down the opening of model databases with such files D3PLOT opens and scans the binout file in a separate thread (effectively in parallel), and results from the binout file only become available when that thread has finished its scan.

NOTE :	As the output frequency of data to each branch of the binout file can be different to the PTF output frequency the outout times might not match exactly. For each PTF state D3PLOT will automatically select the time state in the binout file that is closest to the PTF state time. If the difference betwen the closest binout data is more than 10% of the PTF state interval D3PLOT will generate a warning message.

4.5.8. Files Read Using the D3PLOT <=> T/HIS Link

Files read using the D3PLOT <=> T/HIS link

```
<name>.thf
<name>.xtf
<ascii files>
<binout(LSDA) files>
```

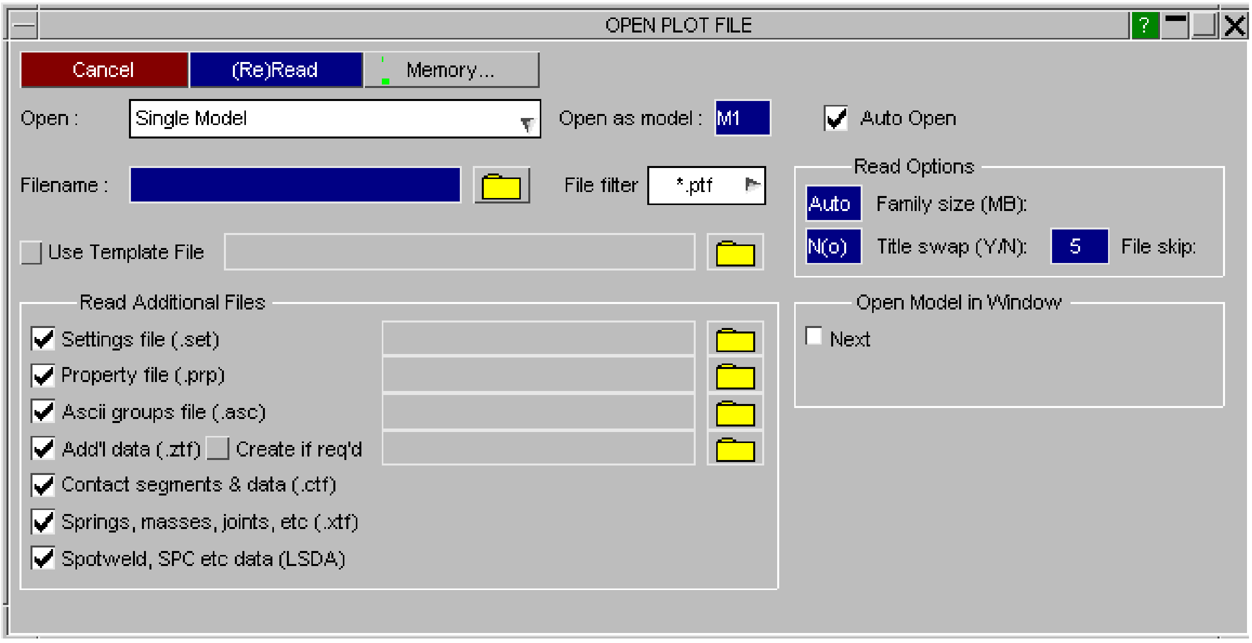
If the optional D3PLOT <=> T/HIS link (see T/HIS the D3PLOT <=> T/HIS Link) is used then D3PLOT gains access to the time-history data embedded in these files. (Ordinarily access to the .XTF file above only extracts geometry data).

The link permits nodes and elements in time-history blocks to be visualised and selected graphically for time-history processing.

4.5.9. Files Required for D3PLOT to Run

Files required for D3PLOT to run

D3PLOT must have a complete state **(.ptf**) file in order to function. If any of the other files (**.xtf** , **.ctf** and **.ztf**) are missing then the entities within them will not be processed.



You can choose whether or not read the **xtf** , **ctf** and **ztf** files by checking the relevant boxes on the front file selection panel.

Here the contact force file (" **Read CTF file** ", etc) has been de-selected, but the other two database files will be read if present.

In D3PLOT 9.3 a **ztf** file can be created automatically if required if the input (.key) deck is available by ticking **Create if req'd** .

Three further, optional files generated from previous D3PLOT runs may be read in: (none of these files is required)

The " **PRP** " file contains model properties, written from the **PROPERTIES** panel.

> This is a model-independent file of element and node properties that can be applied to the current model. "Properties" are colour, transparency, blanking status, etc. See The settings file (.set) stores Programme, not Model, information for more information about this file.

The " **SET** " file containing saved D3PLOT settings written from the **UTILITIES, SETTINGS** panel.

> This is programme-specific data, allowing virtually all the options on the user interface to be saved and restored. For example the number and layout of windows, current data components, etc. See UTILITIES, SETTINGS_FILE for more information about this file.

The " **ASC** " ascii groups file written from the **GROUPS** panel.

> An ascii groups file is a compact and human readable file of group information that can be applied to any model.

4.5.10. Converting Between Binary File Formats

Converting between binary file formats

In D3PLOT 8.0 onwards of D3PLOT conversion of binary file formats to the native type of the processing computer is fully automatic: database files may be generated on one computer and post-processed on another with an incompatible binary format and/or precision. Specifically the conversions between:

- 64 to 32 bit word lengths.
- Cray to IEEE numeric representation.
- Big to little endian organisation

are performed automatically without any input from the user.

4.5.11. Database Filename Syntax

Database filename syntax

D3PLOT supports all the following database filename syntax options from current and previous releases of LS-DYNA:

	"Old" syntax (pre-D3PLOT 6.0)	**"New" syntax, D3PLOT 6.0 onwards**	**Default filenames if none defined**

Database filenames	`<name>.ptf , .p01`
	`<name>.xtf , .x01`
	`<name>.ctf , .c01`
		`<name>.ptf, .ptf01`
		`<name>.xtf, .xtf01`
		`<name>.ctf, .ctf01`
		`<name.ztf>, (n/a)`
			`d3plot, d3plot01`
			`xtfile, xtfile01`
			(none)
			`ztfile , n/a`

Permitted #chars	4 in <name> 3 in <ext>	<name> + <ext> any number less than 80	n/a

Max #family members	99	999	999
Pathname permitted	No	Yes	Yes

4.5.12. Binary File Family Member Size

Binary file family member size

In D3PLOT 8.0 onwards the determination of binary file family member size is automatic by default. D3PLOT takes the actual size of files, rounded up to the nearest Mbyte, as being the effective size for a given family and no further intervention by the user is required.

This can be overridden by giving an explicit size when the file is opened, or subsequently, or by setting the environment variable **`FAM_SIZE`** to an explicit size in MB. For example on a UNIX system:

`setenv FAM_SIZE 1`	(1MByte family, C shell syntax)
`FAM_SIZE=9, export FAM_SIZE`	(9MByte family, Bourne shell syntax)

Any family size is legal, but it is suggested that it be a multiple of 1MByte on single precision machines. On machines generating double precision (64 bit) output files the same numbers (ie 1 and 9 in the examples above) should be used, but the actual file sizes will be 2 and 18 MBytes respectively. This is covered in more detail in Setting the family member size of database files

4.5.13. Permitted Gaps in Family Member Sequences

Permitted gaps in family member sequences

It is possible to skip over gaps in file family member sequences. The "file skip" variable may be set when files are read in, and when the code is running. This is covered in more detail in Handling missing family members .

There are two ways to change the default value. The first method is by setting the D3PLOT preference 'file_skip' in the oa_pref file. For example:

`d3plot*file_skip: 5`	(Skip 5 files)

It is also possible to change this globally on UNIX systems by setting the **`FILE_SKIP`** environment variable. For example:

`setenv FILE_SKIP 5`	(Skip 5 files, C shell syntax)
`FILE_SKIP=9, export FILE_SKIP`	(Skip 9 files, Bourne shell syntax)

On some installations these variables are set globally for all users in the Shell - consult your system manager.

Hint: On UNIX systems you can list all environment variables in the current shell with:

`printenv`

On Windows systems use **Control Panel** , **System** to view and set environment variables.

If they seem to be set correctly, but don't seem to be affecting your process, remember that such variables must be set before the process starts. This is because a child process inherits properties of its parent when it starts, but thereafter is autonomous. You may need to exit and restart the process to make them take effect.

4.6. Other Output Files Processed

Other output files processed

Other results files that D3PLOT can read:

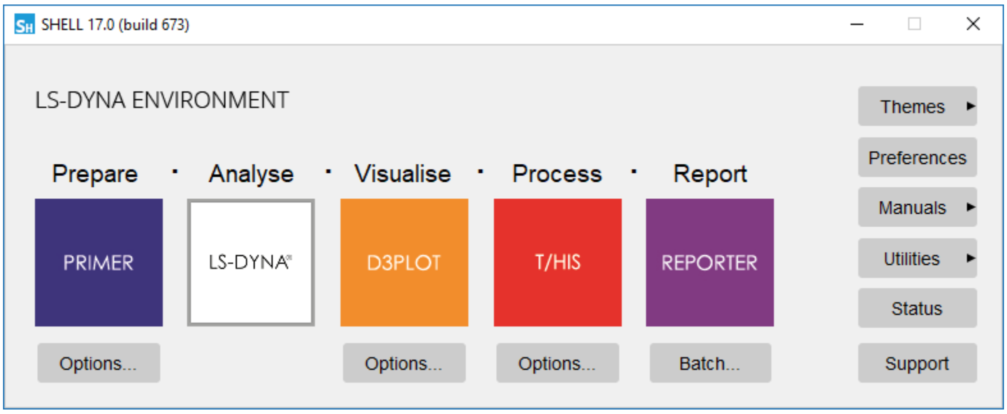
NASTRAN (*.OP2)	Nastran OP2 results file. See APPENDIX F for what is supported.
FEMZIP (*.fz)	LS-DYNA D3PLOT file compressed using FEMZIP (from Frauhofer SCAI).
LS-PREPOST (*.db)	LS-PREPOST database file.

5. Running D3PLOT

5.1. Starting the Code

Starting the code

For users on a device with a window manager D3PLOT is run from the **D3PLOT** button in the Shell:



Users who are running on a device without a window manager should use the `PL` option in the command-line shell.

Users on Windows platforms may associate the filetype ".ptf" with D3PLOT if they wish, so that double-clicking on a .ptf file starts the code. The way to do this is defined in To make .ptf files open in D3PLOT by double-clicking on them .

If your system has been customised locally you may have to use some other command or icon: consult your system manager in this case.

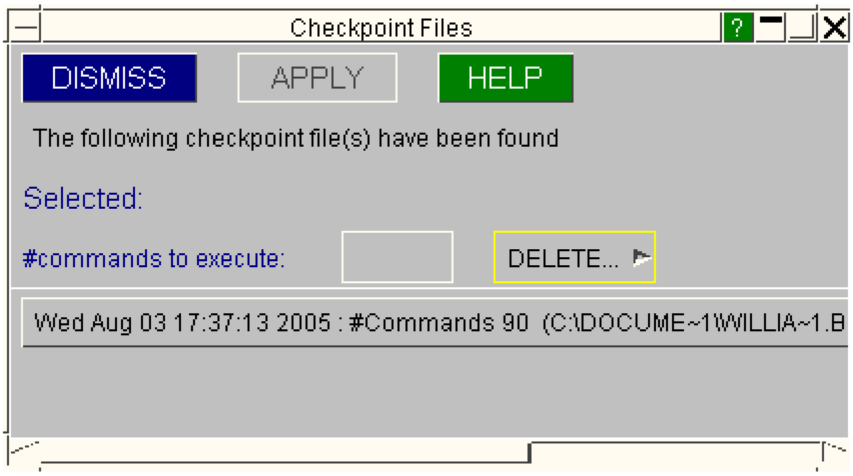
5.1.1. The Checkpoint File Panel

The Checkpoint File panel

A "Checkpoint" file saves all commands and mouse actions during a D3PLOT session. It is deleted when the code terminates normally, but is left on disk if a crash, system failure or some other abnormal termination occurs.

If you see the checkpoint file panel when the code starts this means that a previous D3PLOT run has crashed, and recovery is possible.

Checkpoint files are described in [Checkpoint Files]



5.2. If D3PLOT Will Not Open a Window on Your Display

If D3PLOT will not open a window on your display

If you get a message stating:

Could not open display <hostname>:0

and no window appears, you have failed to make a connection with the X11 server.

(Note that on Linux OpenGL also uses the X11 server, so this section is equally applicable to both X-windows and OpenGL graphics. It does not apply to users running on Windows.)

This is almost certainly because of one or both of the following setup errors:

(1) The **DISPLAY** environment variable has not been set up, or has been set incorrectly, on the "client" machine (where the D3PLOT process is running).

This environment variable tells the X11 window manager on the client machine where to place windows, and it must be set to point to the screen you plan to use. Its generic Unix setup string is:

setenv DISPLAY <hostname>:<display number>	(C shell syntax)

Where <hostname> is your machine's name or internet address, for example:

`setenv DISPLAY :0`	(Default display :0 on this machine)
`setenv DISPLAY tigger:0`	(Default display :0 on machine "tigger")
`setenv DISPLAY 69.23.15.2:0`	(Default display :0, address 69.23.15.2)

You may have to use the raw network address if the machine name has not been added to your **/etc/hosts** file, or possibly the "yellow pages" server hosts file.

(2) The machine on which you are attempting to open the window, the X11 "server", has not been told to accept window manager requests from remote clients.

This is often the case when you are trying to display from a remote machine over a network, and you get a message on the lines of:

```
Xlib: connection to "<hostname>" refused by server
Xlib: Client is not authorised to connect to server
```

In this case go to any window on the server with a Unix prompt and type:

```
xhost +
```

Which tells its window manager to accept requests from any remote client. It will produce a confirmatory message, which will be something like:

```
access control disabled, clients can connect from any host
```

Networked graphics are a complex topic: see Graphics problems for more detailed advice if the remedies here don't work. Alternatively see your system manager, or contact Oasys Ltd for advice and help.

5.3. Client/Server Graphics Using OpenGL

Client/server graphics using OpenGL

It is relatively common to display 2D X-windows graphics from remote hosts on a local server. However it is less well known that exactly the same mechanism can also be used to display 3D graphics under OpenGL.

This can be a very efficient solution to the problem of rapid display of large datasets since the tasks of computing and displaying the graphics and, just as importantly, the memory consumption, are split over two machines. Both machines must be OpenGL compatible for this to work.

To do this:

Set the `DISPLAY` environment variable on the client to point to the server (Starting the code above).

Start D3PLOT in the normal way and read in the model.

Select **OBJECT** display mode (States Box, **ANIM >, DISPLAY MODE, OBJECT**)

This will have the effect of storing all graphics as OpenGL "objects" in the server, making animation and redraw speed extremely fast. However keep an eye on memory consumption of the server: objects may be fast but they use a horrendous amount of memory.

5.4. Command Line Options

Command Line Options

Instead of starting D3PLOT using the Shell it is also possible to start D3PLOT from the command line. Starting D3PLOT from the command line offers a number of advantages.

- Faster start-up is possible by pre-selecting the device type.
- The input filename can be specified and opened automatically.
- Faster start-up is possible by pre-selecting the device type.

Argument format:

<application name> (<arg 1>)...(<arg n>) (<input filename>)

5.4.1. Valid D3PLOT Command Line Arguments

Valid D3PLOT command-line arguments

Selecting a display device	***-d=< device type >***	Device types that are valid are:

<table>
<tr><td></td><td></td><td><table>
<tr><td>opengl</td><td>Default 3D OpenGL graphics, which is standard on both Windows and Linux platforms.</td></tr>
<tr><td>stereo</td><td>Stereo OpenGL on hardware which will support this. It will revert to standard (mono) OpenGL if not available on that platform.</td></tr>
<tr><td>batch</td><td>Default OpenGL, but with the display not shown on the screen, used for batch processing.</td></tr>
<tr><td>tty</td><td>No graphics. Has a very limited functionality and should not be used without advice from Oasys Ltd</td></tr>
</table>
(Older versions of D3PLOT also supported 2d X11 graphics on Linux platforms. This was withdrawn in D3PLOT 14.0, and 2d graphics is no longer supported.)</td></tr>
<tr><td>Flag to start with window maximised (full screen)</td><td>-maximise</td><td>(No argument)</td></tr>
<tr><td>Specifying window placement on a multi-display desktop

By default the top right corner of the desktop is used.

The most common arrangement is two screens</td><td>-placement=< where ></td><td>This option is intended for use where the desktop is spread as a "Single Logical Screen" over multiple monitors.
<table>
<tr><td>< where > values</td><td>Meaning</td></tr>
<tr><td>left</td><td>Left hand monitor</td></tr>
<tr><td>right</td><td>Right hand monitor</td></tr>
<tr><td>top</td><td>Upper monitor</td></tr>
<tr><td>bottom</td><td>Bottom monitor</td></tr>
</table></td></tr>
</table>

side by side, for which "left" and "right" may be used. However "top" and "bottom" are also available for the case of two screens one above the other, and the options may be concatenated for a 2x2 display. These options can be combined with -maximise to fill the relevant screen. Users on Windows platforms where tools such as NVidia's "NView" are available may find that it is better to leave window placement to that tool, so that D3PLOT's windows behave in a fashion consistent with other application windows.		The above may be concatenated for a 2x2 display, for example	
		`top_left`	Top left monitor
		`bottom_right`	Bottom right monitor

Command file name Flag to exit when command file run complete if desired	**-cf=< *filename* >** **-exit**	A valid command file name (usually *.tcf) (No argument)
JavaScript	**-js=< *filename* >**	Any valid D3PLOT JavaScript file
JavaScript Arguments	**-js_arg=< *argument* >**	Any valid string The arguments can be accessed in the script by using the global **arguments** array. Multiple arguments can be given to a script by using more than one `-js_arg` command line argument.
Checkpoint file to replay Number of lines to execute in checkpoint file	**-replay=< *filename* >** **-rlines= *<nnnn>***	A valid D3PLOT ckeckpoint file (usually **cp_d3plot** *nnnn*) Where < *nnnn* > is a positive integer
Alternate "start in" directory (redefines current working directory)	**-start_in=< *pathname* >**	A valid directory (eg c:\my_files, /data/my_files)
Optional "project"	**-pcwd=< *pathname* >**	A valid directory (eg c:\proj_files, /data/proj_files)

working directory. This specifies an alternate initial location for view, cut-section, group, settings and external data files. Useful if the directory containing analysis data is read-only so that these files have to be located elsewhere		
Specifying a custom "oa_pref" file. This causes an extra, optional "oa_pref" file to be read.	**-pref=<filename>**	<filename> must be a valid "oa_pref" file. If it has no path prefixed, the file is assumed to be in the OA_INSTALL directory. Any legal filename may be used.
Specify a file that contains commands to create a cutdown version of the ptf file (see [WRITE] CUTDOWN D3PLOT/PTF File).	**-ptfcut=<filename>**	

<table>
<tr>
<td>As above but allows creation of cutdown ptf files (see Disk Format of Binary Database Files) in true batch mode (-d=tty) using easily readable/editable options. Subsidiary options may be specified either on the command line or fed in from a file.</td>
<td>-ptfcut_batch <=filename></td>
<td>The following data components are accessible using this flag:
<table>
<tr><th>Flag</th><th>Component</th></tr>
<tr><td>-acc</td><td>Acceleration</td></tr>
<tr><td>-plastic</td><td>Plastic strain</td></tr>
<tr><td>-strain</td><td>Strain tensor</td></tr>
<tr><td>-stress</td><td>Stress tensor</td></tr>
<tr><td>-vel</td><td>Velocity</td></tr>
<tr><td>-von_stress (reordered format only)</td><td>Von mises stress</td></tr>
<tr><td>-von_strain (reordered format only)</td><td>Von mises strain</td></tr>
</table>
The following output formats are available:
<table>
<tr><th>Flag</th><th>Format</th></tr>
<tr><td>-reorder</td><td>Reordered</td></tr>
<tr><td><none></td><td>Original</td></tr>
</table>
Specific states may be selected as follows:
<table>
<tr><th>Flag</th><th>Meaning</th></tr>
<tr><td>-states=odd</td><td>All odd states</td></tr>
<tr><td>-states=even</td><td>All even states</td></tr>
<tr><td>-states=start,end,optional_inc</td><td>States between 'start' and 'end' with a step size of 'optional_inc'; end can</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td rowspan="2"></td><td rowspan="2"></td><td></td><td>be "last" (minus quoes)</td></tr>
<tr><td>-states=all

<none></td><td>All states selected</td></tr>
<tr><td>...continued...</td><td></td><td colspan="2">Two new methods of generating cutdown models are available:

Method 1 involves feeding all relevant options on the command line. This can only produce one cutdown model. Eg:

-ptfcut_batch -reorder <-vel/-acc/-plastic/-stress/ -strain/-von_stress/-von_strain> -states=<[start, end, optional_inc]/odd/even> -o=<output> <input>

Method 2 involves specifying options in a file and invoking that using the -ptfcut_batch= flag. This method can be used to generate multiple cutdown models, each with its own components, formats and states

-ptfcut_batch=<options_file> <input>

Example options_file:

-reorder -vel -states=odd -o=vel.ptf
-acc -o=acc.ptf

There is currently no option to select specific parts for the cutdown model. All parts are processed by default.</td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td>Specify a GLB output file for 3D Export.</td><td>-glb=<file></td><td colspan="2">Generate a compressed GLB file</td></tr>
<tr><td></td><td>-glb_uncompressed=<file></td><td colspan="2">Generate an uncompressed GLB file</td></tr>
<tr><td></td><td colspan="3">The following options can be used with this flag to control the component and states captured in the 3D Viewer Export.</td></tr>
</table>

	`-components=A,B,C`	A comma separated list of components, where each component generates a separate GLB file.
		
		The component names can either be the JavaScript Data component names or the D3PLOT button label names.
		
		`-components=svon`
		`-components=X_DIRECT_STRESS`
		
		Each component can be followed by a surface (top/middle/bottom/max/min/mag) or a specific layer number if required.
		
		`-components=svon@top`
		`-components=X_DIRECT_STRESS@1`
		
		A special component "none" can also be specified in the list to generate a Shaded plot.
		
		`-components=svon@top,X_DIRECT_STRESS@1,none,vm`
		
		would generate 4 output files
		
		• Von Mises Stress (Top Surface)
		• X Direct Stress (Layer 1)
		• Shaded Plot
		• Velocity Magnitude

	`-frame_rate=N`	Specifies the default playback frame rate for D3PLOT Viewer (www.d3plotviewer.com).
	`-states=N`	Select single state N
	`-states=odd`	All odd states
	`-states=even`	All even states
	`-states=start,end,inc`	States between 'start' and 'end' with a step size of 'inc'; end can be "last" (minus quotes), inc is optional

	-states=all **<none>**	**All states selected. This is the default, unless D3PLOT reads a settings file along with the model, in which case the default is to export a single state (the current state specified by the settings file).**
Specify a D3PLOT template file that contains information on which models are loaded into each window and any model offsets for each window (see Template File).	**-tpl=<filename>**	
Specify the name of a model database file to open (see Select Models From Database).	**-mdb=<filename>**	
Specify an alternate location for a ZTF file to read. This option can be useful if PRIMER is unable to create a ZTF file in the same location as the D3PLOT PTF files (see Open a Single Model).	**-ztf=<filename>**	
Do not read a ZTF file.	**-no_ztf**	

Specify an alternate location for a D3PLOT settings file. By default D3PLOT will look in the directory containing the PTF files for a setting file to read (see Open a Single Model).	**-set=<filename>**	
Do not read a D3PLOT settings file.	**-no_set**	
Specify an alternate location for a D3PLOT properties file. By default D3PLOT will look in the directory containing the PTF files for a properties file to read (see Open a Single Model).	**-prop=<filename>**	
Do not read a D3PLOT properties file.	**-no_prop**	
Specify an alternate location for a D3PLOT groups file. By default D3PLOT will	**-group=<filename>**	

look in the directory containing the PTF files for a groups file to read (see Open a Single Model).		
Do not read a D3PLOT groups file.	**-no_group**	Using this option will prevent D3PLOT from reading a human-readable ***ASCII*** groups file (group *nnn* .asc) but it will still read any cached ***binary*** groups file (*jobname* .grp) that is left over from an earlier D3PLOT session which used groups in this model. For more information see the GROUPS section of this manual.
Specify a file containing a list of models for D3PLOT to automatically open.	**-ml=<filename>**	The model list file should contain the full pathname of one file from each model that D3PLOT should open. Each file should be on a separate line and it should be the first item on each line. By default each model will be read into Window 1, but you can specify which windows a model is read into by specify a bitwise encoded number after the model name (W1=1, W2=2, W3=4, W4=8, etc.) e.g. if you read in 4 models with the following file: *model1.ptf 1* *model2.ptf 2* *model3.ptf 4* *model4.ptf 3* model1.ptf would go into W1, model2.ptf into W2, model3.ptf into W3 and

		model4.ptf into W1 and W2. Contact Oasys Ltd if you need further explanation.
Do not read a file with interface force segments and data.	**-no_ifs**	
Do not read a file with springs, masses, joints etc.	**-no_xtf**	
Do not read a spotweld, SPC etc. data file.	**-no_lsda**	
Run D3PLOT in "batch" mode where the main application window is not displayed on the screen. For this option to work you must also specify a command file " **-cf=filename** " and the name of the PTF file to open. This option will automatically set " **-exit** " so that D3PLOT terminates after playing	**-batch**	

the command file.		
Redirect output from the console window to a file on Windows. To redirect output on Unix/Linux use the shell redirection options (typically > for <stdout>, & for <stderr>)	**-eo=<filename>** **-eo** **-eo=default**	-eo=<filename> is designed for the user to suppress the console and redirect logfile output to the specified filename. In order to permit multiple sessions to coexist on the same machine the process id will be appended to the <name> part of the filename to give <name>_pid.<ext>. If plain "-eo" or "-eo=default" are found then filename generation is automatic, and the first valid of: **%TEMP%\this_log_<pid>.txt** **%TMP%\this_log_<pid>.txt** **%HOMESHARE%\this_log_<pid>.txt** **%USERPROFILE%\this_log_<pid>.txt** will be used.
Read/Write checkpoint files Start writing the checkpoint files upon D3PLOT startup Read checkpoint files and Show checkpoint playback panel upon D3PLOT startup. Directory path to write checkpoint files	**write_checkpoint_files=< *TRUE/FALSE* >** **show_checkpoint_files=< *TRUE/FALSE* >** **checkpoint_dir=< *directory* >**	TRUE/FALSE, turn on/off the writing of the checkpoint files (default is FALSE) TRUE/FALSE, turn off the initial checkpoint files panel (default is FALSE) If the writing of the checkpoint files is OFF, the reading will also be OFF < *directory* > must be a valid directory name on your system. If the value is <none> then the checkpoint files are not recorded for the D3PLOT session.

<table>
<tr><td>Run D3PLOT without the console window.</td><td>-noconsole</td><td colspan="2">Windows only.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td rowspan="3">Input database filename.

(The extra time history (.xtf) and contact force (.ctf) databases are also opened if present.)</td><td rowspan="3">< filename >

eg run_1.ptf</td><td colspan="2">A valid input file type:</td></tr>
<tr><td>name.ptf</td><td>(Complete state file)</td></tr>
<tr><td>d3plot</td><td>(ditto)</td></tr>
</table>

Some examples for D3PLOT might be:

pathname/d3plot18.exe -maximise run_2.ptf	(Use full -screen, open file run_2.ptf)

Note that no spaces should be left in the syntax <arg>=<value>.

For example:	" **-eo = default** " is illegal.
Correct syntax is:	" **-eo=default** "

5.4.2. WINDOWS (PCs)

WINDOWS (PCs)

Command-line arguments on Windows

It is possible to define command-line arguments under Windows: either directly when running an application from a MS-DOS prompt, or by defining "action" arguments when configuring a shortcut (see Appendix D for more details). However this is unusual, and it is suggested that you seek advice from Oasys Ltd if you are not sure how to do this.

Click here for the next section

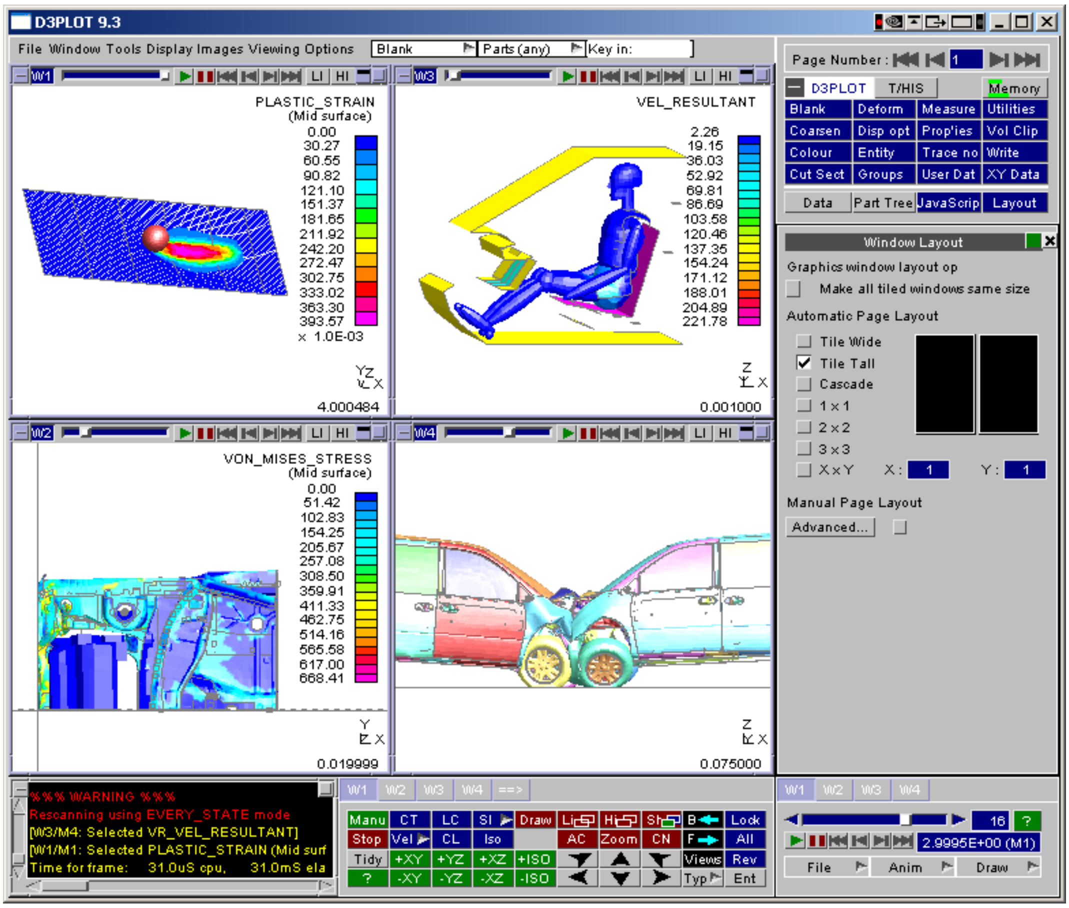
5.5. Multiple Windows and Models

Multiple Windows and Models

From D3PLOT 9.3 (Nov 2008) onwards, D3PLOT supports the following permutation of multiple windows and models.

- Up to 32 windows may be defined.
- Up to 32 models may be current in memory.
- Any permutation of model(s) may appear in any window(s), subject to the limit of 100 "instances" of window/model combinations.
- Windows may be arranged on up to 32 "pages".

This example shows four separate models in four separate windows.



This example also shows how each window can have totally separate attributes: display mode, state number, view, background colour, etc. These can be controlled separately or collectively by using the "tabs" on each menu panel.

Where a window contains multiple models all that models in that window are given the same attributes (component, surface, etc); however it is possible to distinguish between models by:

- Separating them artificially in space
- Giving them different colours
- Drawing them in different modes (wireframe, shaded, etc)

5.5.1. FILE > Menu: Opening, Closing and Rereading Models

FILE > Menu: Opening, Closing and Rereading Models

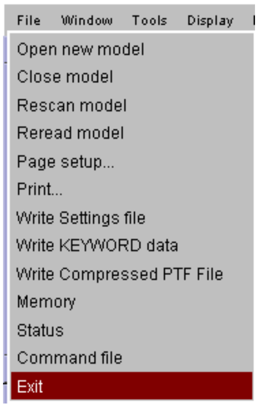
The **OPEN NEW MODEL** command maps the standard input file selection panel (see Reading Results)

Each new model will be opened in a new window. Up to 20 models may be held concurrently in the database, but you should note that they all compete for the same memory resources in the computer and that performance may be badly affected if you try to read in too much data.

When you **CLOSE** a model all windows that display it only will also be closed, and the model's storage deleted from memory. You are warned before this happens.

RESCAN should be used when an analysis is still running and you want to search for more states.

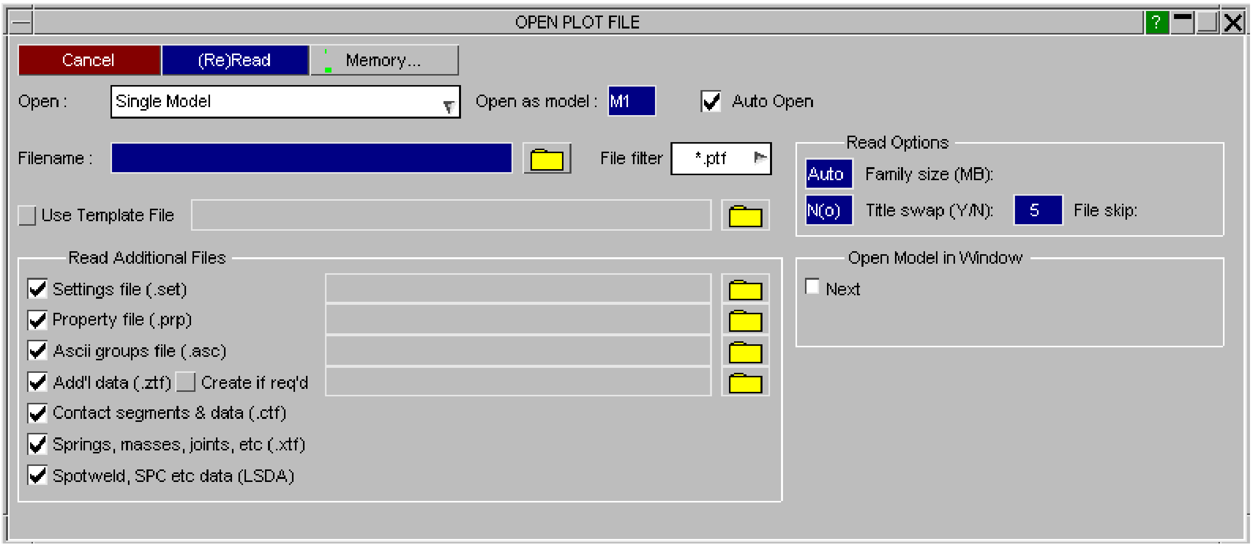
REREAD is equivalent to a Close/Reopen sequence: it completely rereads a model from scratch, and should be used if a model has been rerun. It should also be used when an adaptively remeshed file family has been extended

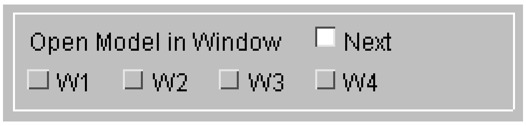


5.5.1.1. Choosing Which Window(s) to Read a New Model Into

Choosing which window(s) to read a new model into

The first model opened is always read into window 1, but models after that may choose which windows they will become active in.





By default they are read into a new window, the "next" one, but you can select any other window(s) as destinations using the "In Window" buttons.

Any permutation of buttons may be selected, and the new model will become active in those windows.

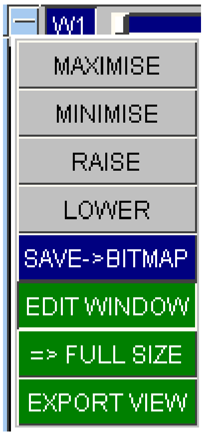
See Open a Single Model for more details

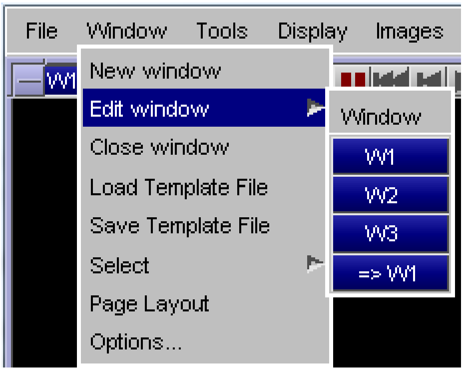
You can subsequently activate and de-activate models in any window at will by using the **EDIT WINDOW** popup menu described below. This also allows you to separate models, set their drawing mode and also their colour.

5.5.1.1.1. EDIT WINDOW Changing "Model in Window" Attributes

EDIT WINDOW Changing "Model in Window" attributes

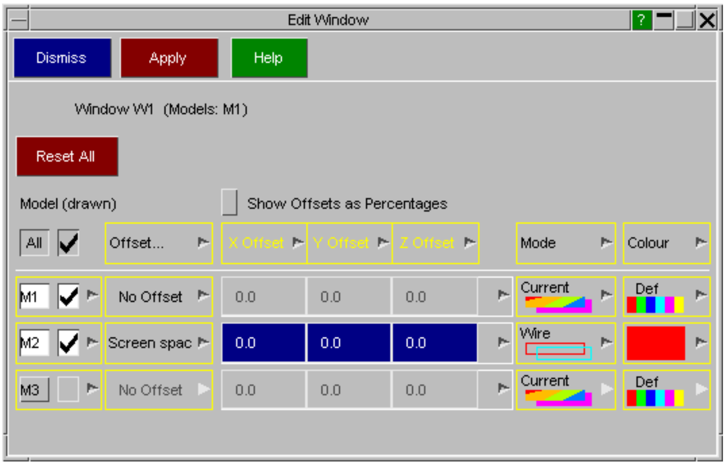
The models active in a window, and some of their attributes, can be changed at any time using the **EDIT WINDOW** option in the popup linked to the top left `[-]` button, or from the window menu on the top bar.





This maps the panel to the right, here for window W1.

In this example 2 out of the 3 models are active in this window. Model M2 has been offset in screen space and is drawn wireframe in red.



MODEL turning models on and off

Simply toggle each model on and off. To see more attributes of the model right click to get its title and filename.

Note that if a model is not active in any window then it will be deleted from the database, you are warned and made to confirm this before it happens.

OFFSET... adding an artifical offset to models in windows

Multiple models in a window often overlay one another, and it can be useful to separate them. Right-click on the relevant OFFSET... button to map this panel, then choose offsets in one of:

- `Model Space` . Shifts the model in its own space system.

- `Screen Space` . Shifts the model in the plane of the screen.

You can visualise the difference between these two by considering how two dancers on a stage would react to rotation by 180 degrees when separated. In model space they would effectively swap sides of the stage; whereas in screen space they would each pirouette about their own toes, staying in the same positions.

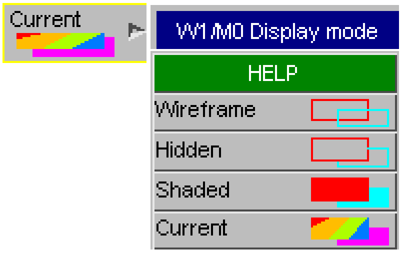
After selecting the offset mode the X, Y, and Z offset values can be entered into the text boxes.

The offset values can all be reset to zero using the popup menu.

The offsets can either be entered in model units or they can be entered as a %age of the model dimensons. Toggeling this option on and off will automatically convert any offsets that have been entered between %ages and model units.

MODE changing the display mode of a model in a window

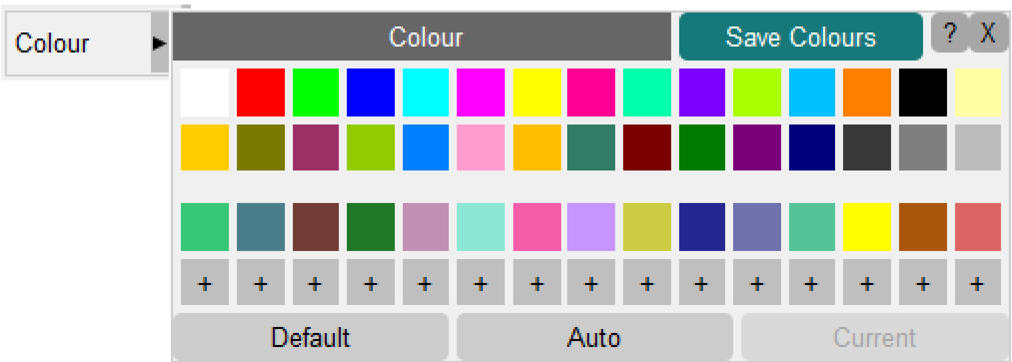
Normally models are displayed in the current mode of the window, whatever that may be, but you can restrict them to:



WIREFRAME	No shading, hidden-surface removal or contouring; edges only.
HIDDEN	No shading or contouring, and edges only, but with hidden surface removal enabled
SHADED	Shading, but no contouring. Both edges and and lit surfaces displayed
CURRENT	Whatever the current display mode is.

The actual display mode used for a model will be `min(current mode, mode selected here)` . In other words selecting SHADED here will only produce a WIREFRAME plot if the current mode is only WIREFRAME (eg **LI**)

COLOUR setting a constant colour for a model in a window



Normally a window in a model will be drawn using its normal colours, which are properties of the model itself.

You can override this by setting a constant colour for the model which will be used instead, which can be useful for distinguishing between two similar models that have been overlaid.

To return to normal colouring for the model use the **Default** button.

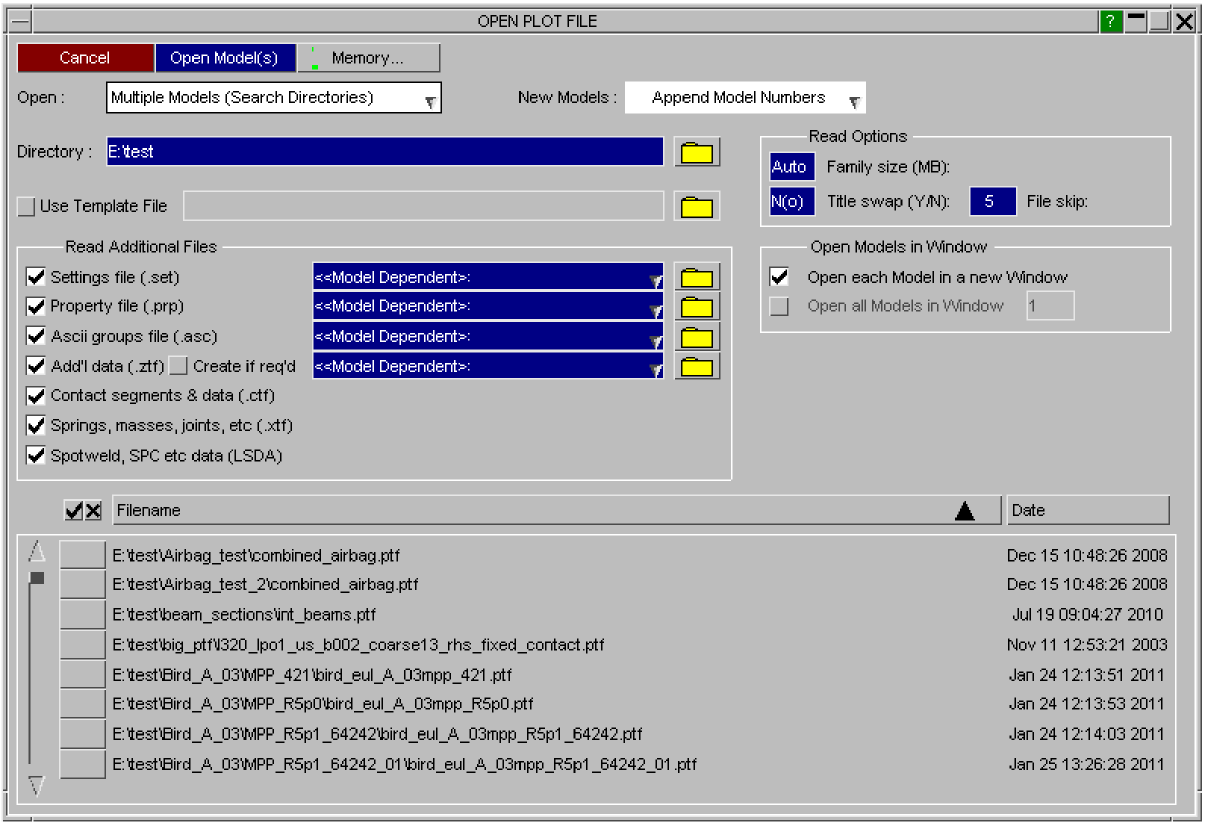
The **Auto** button will automatically assign a colour to each model.

(Note that you can achieve the same effect by changing the colour of elements in the model using the **PROPs** or **COLOUR** panels, but this will apply to all windows in which the model appears, whereas this option only affects the display of the model in this window.)

5.5.1.2. Opening Multiple Models in a Directory Tree

Opening Multiple Models in a Directory Tree

From D3PLOT 9.3 onwards it is possible to open up to 32 models simultaneously from a directory tree.

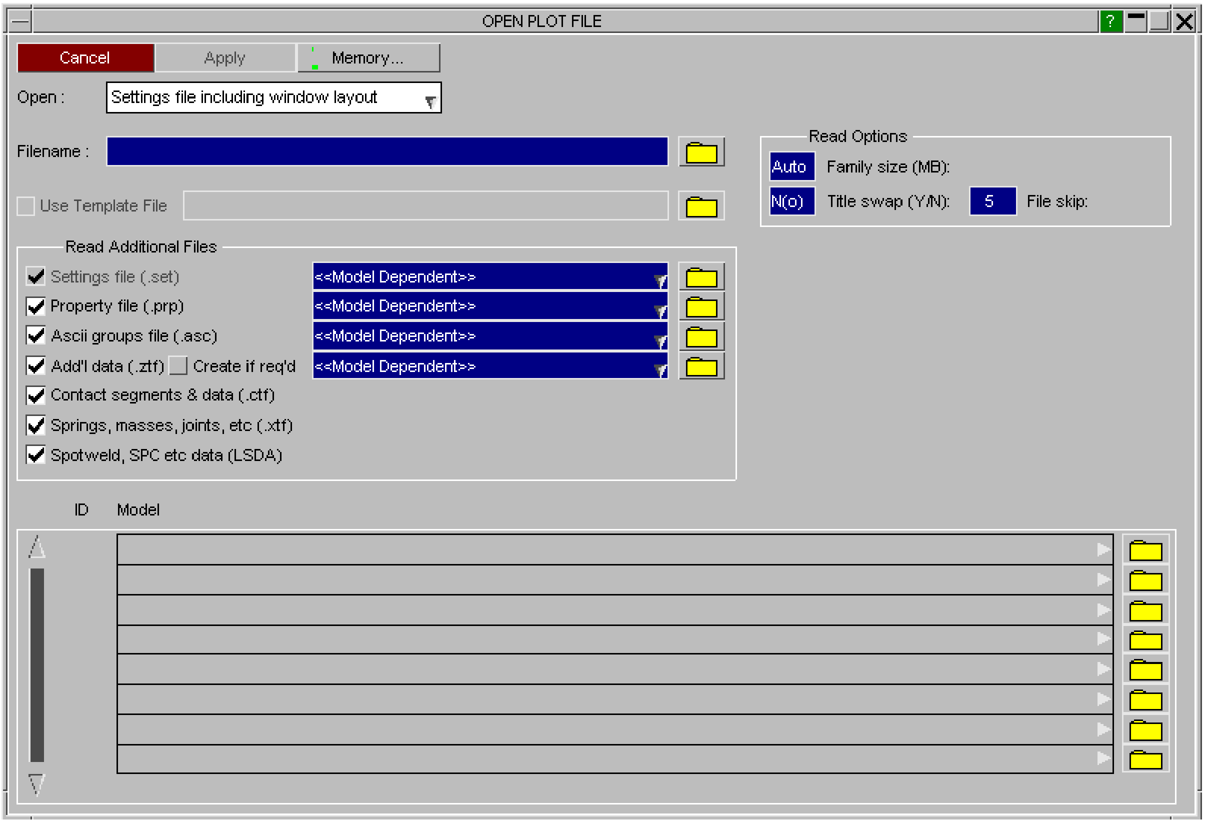


See Search Directories Recursively for more details.

5.5.1.3. Opening Multiple Models Using a Settings File

Opening Multiple Models using a Settings File

From D3PLOT 9.3 onwards it is also possible to open up to 32 individually chosen models with the attributes and layout specified in a Settings file.



See Settings File Including Window Layout for more details.

5.5.2. WINDOW > Menu: Window Management

WINDOW > Menu: Window management

NEW WINDOW creates new windows. If there is more than one model in memory you have to choose the model to be placed in the new window. The newly created window will always be numbered as the next free one in the labelling sequence, and positioned in its default " layout ".

EDIT WINDOW raises the window content editing panel described above once you have selected which window you want to operate on.

CLOSE WINDOW allows you to close any permutation of windows. If you close all the windows used by a model then that model is also deleted from memory. When a window in the middle of a sequence is closed the remaining windows above it are renumbered downwards so that there are no gaps in the sequence.

SELECT > is the "global" window tab selector. This topic is covered in more detail below.

LAYOUT... controls how multiple windows are organised on the screen.

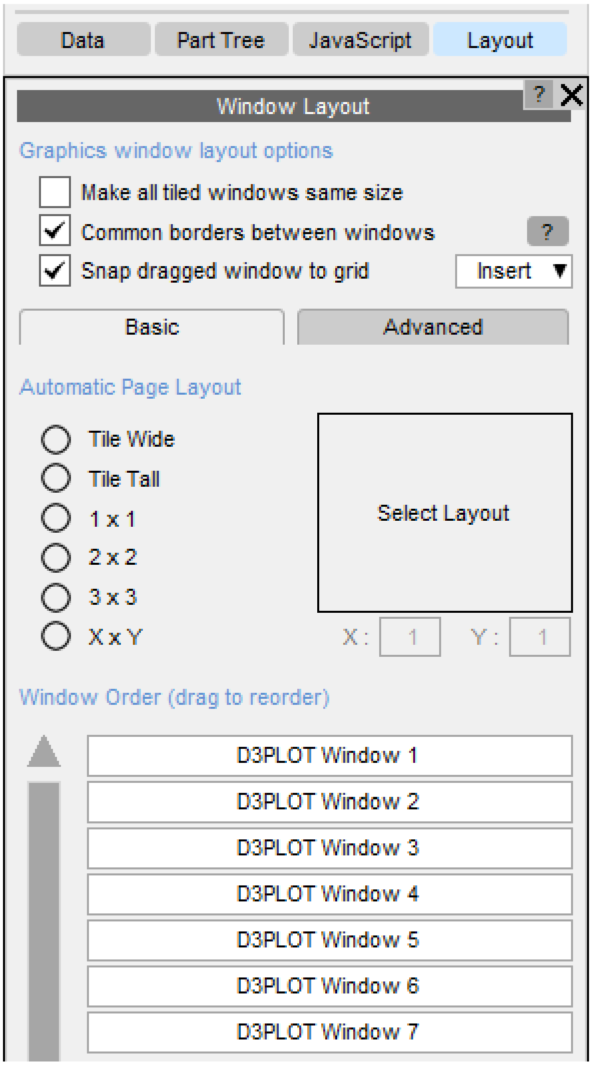
OPTIONS... controls further aspects of window management and display.

5.5.3. Layout

Layout

Windows can be laid out in a number of different formats and can be organised into Pages.

From D3PLOT and T/HIS 19.0, the Window Layout menu is split into separate **Basic** and **Advanced** modes.



5.5.4. Basic Mode

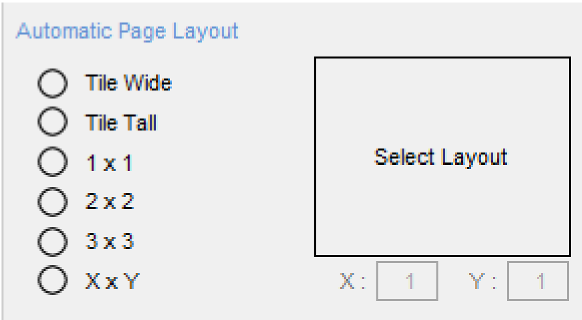
Basic Mode

In Basic mode, the menu can be used to select a page layout that is automatically applied to all of the pages.

Automatic Page Layout

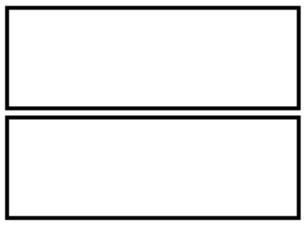
If an Automatic page layout is used and the layout is set to **Tile Wide** or **Tile Tall** then all windows are automatically added to page 1.

In all other layouts, Windows are automatically added to pages and as many pages as needed are created to hold all the windows.



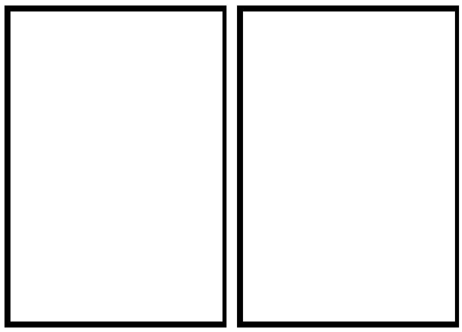
Tile Wide

All of the windows are positioned on a single page.



Tile Tall

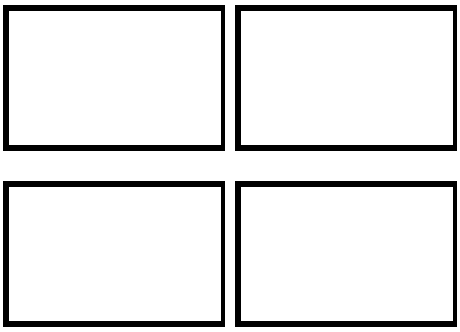
All of the windows are positioned on a single page.



1 x 1

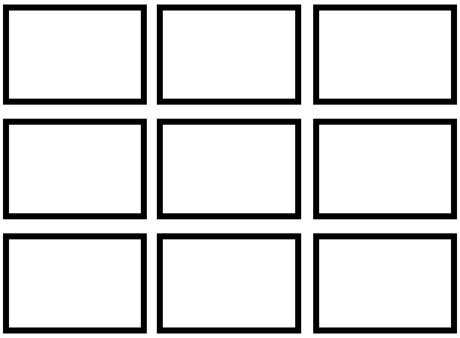
Each window is positioned on its own page.

2 x 2



Windows are arranged in a 2 by 2 grid. If there are more than 4 windows, then windows 1 to 4 are positioned on page 1, windows 5 to 8 on page 2, etc.

3 x 3

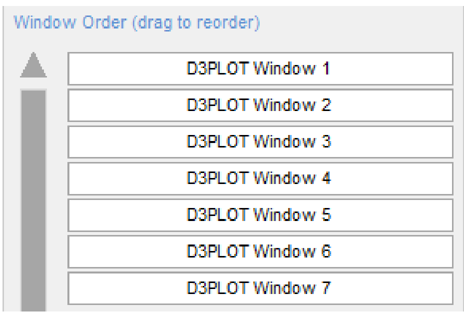


Windows are arranged in a 3 by 3 grid. If there are more than 9 windows then windows 1 to 9 are positioned on page 1, windows 10 to 18 on page 2, etc.

X x Y

Windows are arranged in a X by Y grid.

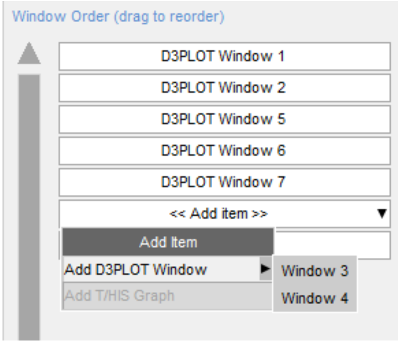
Window Order



By default, Windows are added to pages in the order they are created.

The order of Windows can be changed by clicking on a row and dragging it up or down the list to a new position.

Any Windows that have been dragged out onto the desktop are removed from the list (Windows 3 and 4 in the example on the right).



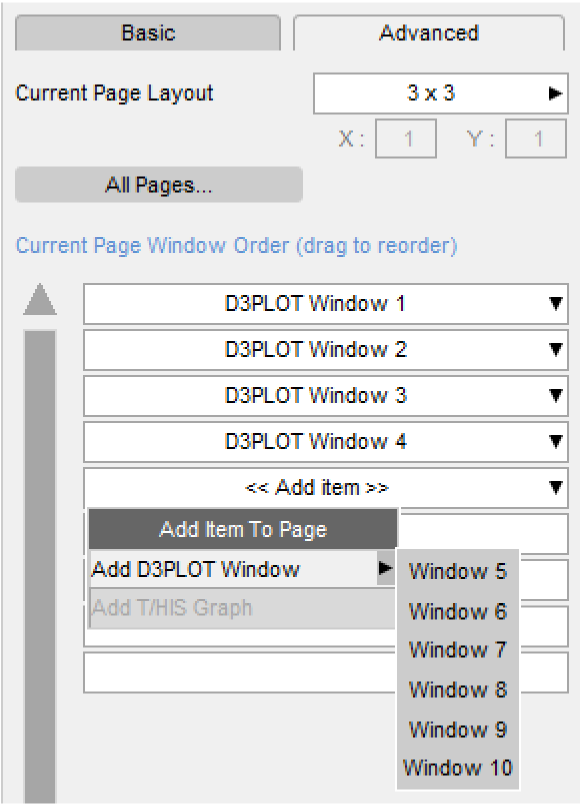
If Windows are on the desktop, the menu will display additional rows that can be used to add the windows back into the list so that they are displayed on a page again.

Right-clicking on an **<< Add item >>** row will display a popup menu that can be used to select any Windows currently on the desktop.

In a Linked D3PLOT T/HIS session, this menu will also display any T/HIS graphs that are currently docked inside D3PLOT windows so that they can be added back onto a page.

5.5.5. Advanced Mode

Advanced Mode



Advanced mode can be used to give more control over which windows appear on which page. Unlike in Basic mode, a window can appear on more than one page.

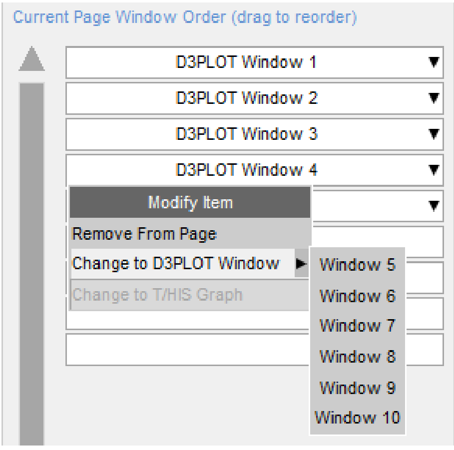
Advanced mode works in a similar way to Basic mode except that it controls the settings for the current page only.

In Advanced mode, the layout and content of each page can be set for that page, and the order of the items displayed on each page can also be controlled by clicking on an item and dragging it up or down to a new position.

Right-clicking on **<< Add item >>** will display a popup menu that can be used to select any Window that is not currently on the page.

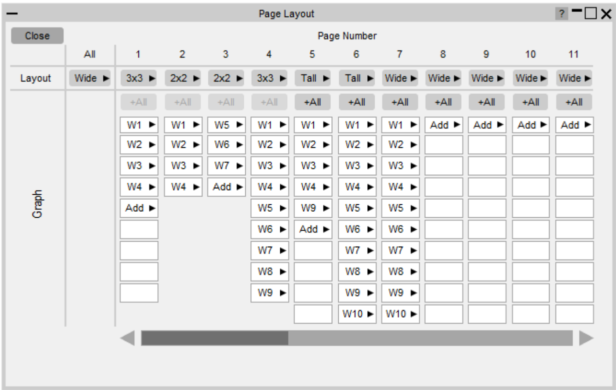
Right-clicking on a row containing a Window will display a popup menu that can be used to remove the Window from the current page.

Alternatively, the same popup can also be used to change an item to a different D3PLOT Window or T/HIS Graph that is not already on the current page.



All Pages menu

In Advanced mode, the **All Pages** button can be used to display a separate menu that shows the layout and contents of all pages:



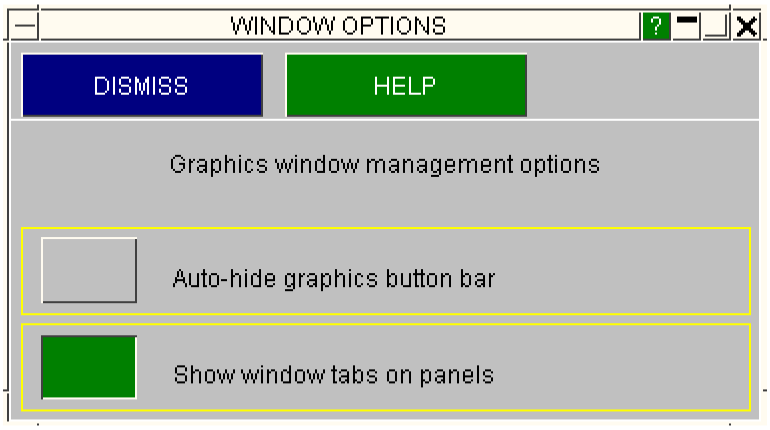
This Page Layout menu can be used to select which windows appear on each page. Each window can appear on more than one page.

The options to reorder or change the contents of each page are similar to those in the Window Order section of the Layout menu:

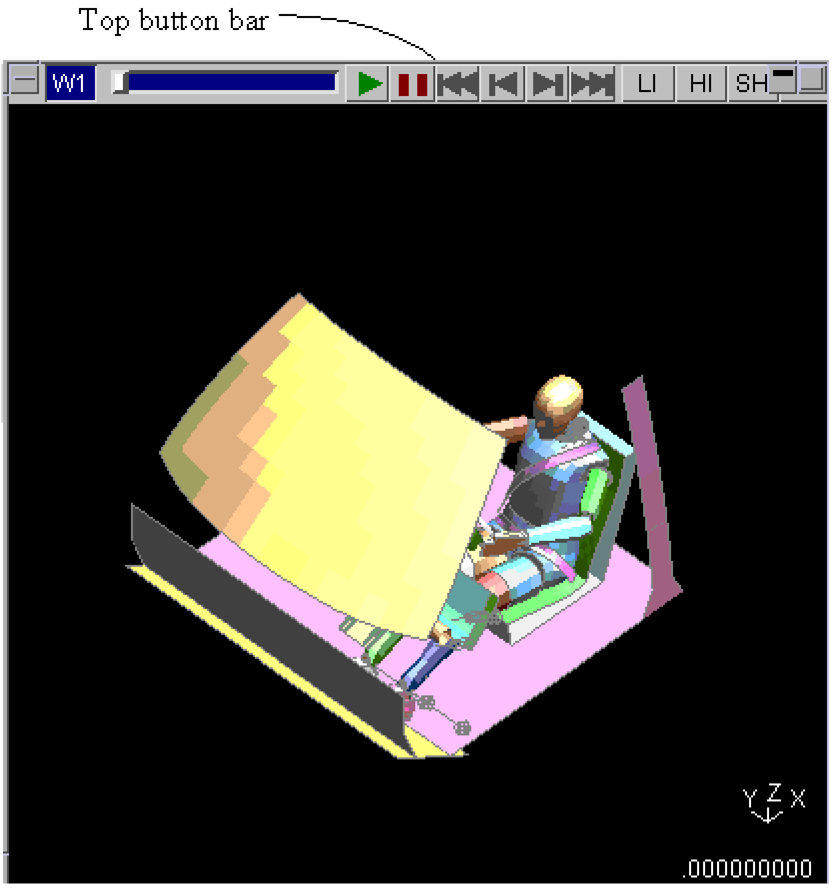
- Drag the buttons in each column up and down to reorder windows and graphs on a page
- Use the popup menus to edit page contents

5.5.6. Window Options...

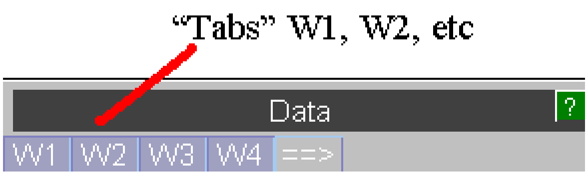
Window **Options...**



" **Auto-hide graphics button bar** " automatically maps the button bar at the top of each window when the cursor enters that window, and hides it again when the cursor leaves. This can be useful when you have many windows, or a small display, as it maximises the amount of space available for graphics. By default auto-hide is off, and the button bar is permanently displayed in all graphics windows.



" **Show window tabs on panels** " controls whether or not the W1, W2, ... "tabs" for multiple graphics windows are displayed at the top of menu panels. If these are suppressed you will not be able to control the application of commands to windows on a per-panel basis. Window "tabs" are discussed in more detail below.



5.5.7. Controlling How Commands Apply to Windows

Controlling how commands apply to Windows

When there is only one graphics window current then there is no ambiguity about where commands issued in menu panels apply, but once two or more windows are current the situation becomes more complicated. For example you may want to contour X stress in window 1, but Y stress in window 2.

To handle this problem D3PLOT automatically adds "tabs" (**W1** , **W2** , ...) for each graphics window to most menu panels when two or more graphics windows are present.

Using Wn tabs on menu panels

Commands in menu panels will only apply to those graphics windows for which the **W *n*** tab buttons are pressed. In the example above all four windows will be affected when commands are given in this panel. If a tab buttons is deselected then subsequent commands in this panel will not apply to that graphics window.

Some further rules apply:

- Each top level menu panel tabs selection is independent. Deselecting a tab button in one panel will not affect any other top level panels on the display.
- Selections propagate downwards to newly mapped children. For example if the **Component** panel is invoked from the **Current Operations** one it will initially inherit its parent's tab selection. However it is not limited to this and can subsequently be changed.
- The current status of a menu panel is influenced by its tabs setting as follows:
    - Where only one status word is displayable (eg component, shell surface) the first active window's value is shown.
    - Where status button is mixed (eg ON in window 1, OFF in window 2) then "ON" will be shown, but on a grey rather than coloured background.

- If no tabs are active then the whole panel will be de-activated

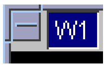
Propagating settings for a panel: the ==> button

Sometimes you may want to propagate settings from one window to others. This can be done in a limited way on a "per menu panel" basis using the **==>** button, which:

- Takes the settings for the first active window in that panel;
- Copies them to all subsequent active windows.

This is mostly commonly used in the View Control panel to make all windows have the same view as W1.

Using the Wn buttons on graphics windows



Here the **W1** button for window #1 is shown.

Each graphics window has its number given in the **W *n*** button at its top left corner. which can be used to toggle on/off that window's tabs in all current menu panels.

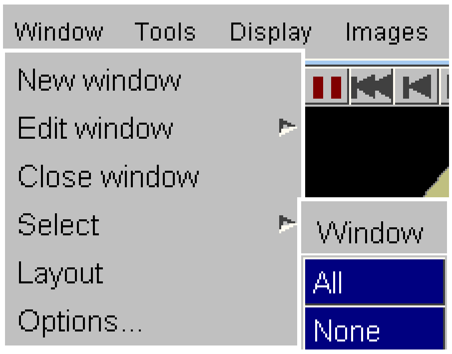
When the **W *n*** button is toggled OFF:

- The border round the graphics window changes from light blue to grey
- The **W *n*** tab in all active menu panels is toggled off, and the panels updated.

When it is toggled on again the opposite occurs.

This is simply a quick way of changing the tab status of all active menu panels: the window is not de-activated in any way, it can still be drawn in, and the local tab for this window can still be changed on any menu panel.

WINDOW > Select ... (De-)Selecting all windows



It is possible to select and deselect all windows.

This is equivalent of toggling the **W *n*** buttons on all graphics windows on (**ALL**) or off (**NONE**), and all menu panels will be affected.

5.5.8. Which Settings are "Per Model" Not "Per Window"?

Which settings are "per model", not "per window"?

It will be obvious that different windows may present different views of a model, or contour different data components; but windows on the same model are not totally independent of one another. Storing detailed attributes for every item in a model on a per window basis would be wasteful of memory, and would also require some very complicated panels to provide detailed feedback to the user.

Therefore "properties" of a model, which can be stored in a properties file, are stored on a per-model rather than a per-window basis. This means:

- Blanking status; as controlled in the **BLANK** panel.
- Colour, transparency, brightness and shininess; as controlled in the **PROPs** and/or **COLOUR** panels.
- Exploded part status; as controlled in the **DEFORM** panel.

"Properties" files are described in more detail under Properties.

If it is necessary to have multiple windows with different blanking or element attribute properties then you will have to read the same model in twice. D3PLOT will treat this as two totally separate models, and you will be able to set different attributes, however it will double your memory usage.

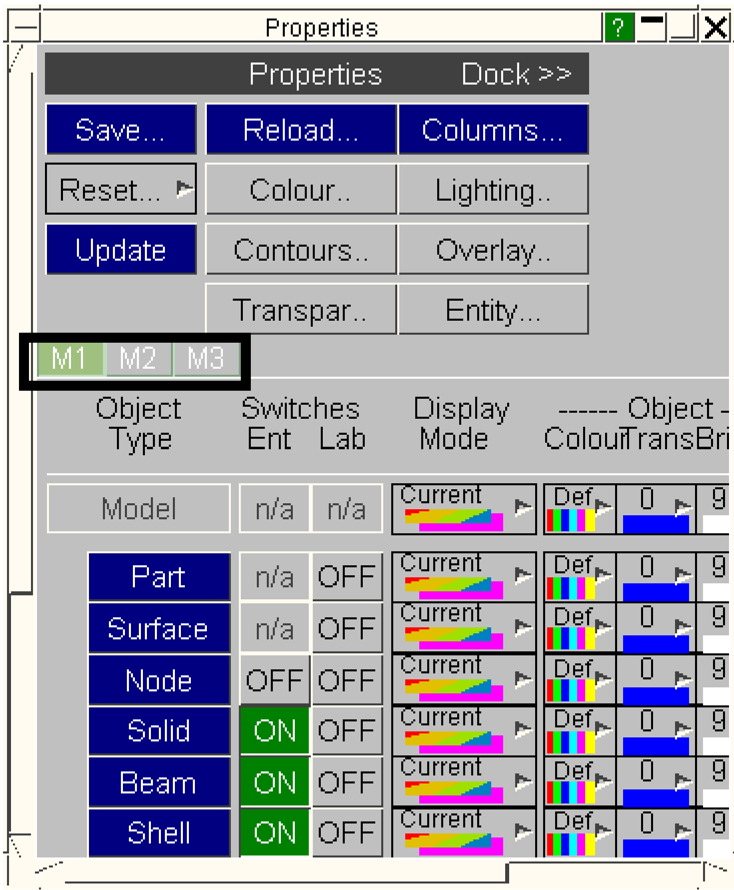
Swapping between models in model-specific panels

Model-specific panels can only operate on one model at a time, but you can swap between models at any time using **M1, M2, ...** tabs.

These function just like window tabs (**W1,** ...) except that:

- They are green rather than blue
- Only one tab may be active at a time

If you only have one model in memory then these tabs are not drawn.



5.5.9. Handling Multiple Models and Multiple Windows

Handling multiple models and multiple windows

Where multiple windows exist on a single model there is no possibility for conflicts between windows when commands are applied. However when multiple models have been read in lots of potential problems arise, to name just a few:

- Models may have different numbers of states, at different times.
- Some models may contain data components that don't appear in others
- Models may contain different types of element

D3PLOT has to protect itself against internal conflicts arising from attempts to impose invalid settings on windows, and also has to allow the user to manipulate multiple model settings in a simple way. So the following rules apply where multiple, dissimilar models are current:

- Where a panel would apply an invalid setting to a model no action is taken. For example turning on spring labels in a model that contains no springs will have no effect. Note that the feedback in a menu panel is generally for the first active window, so to see the status of the 2nd or subsequent models it may be necessary to adjust the tabs so that the model in question is the "first active" one for that panel.

- The **State Number** slider will show the highest state number of all the current models. If it is set to a state that doesn't exist in a given model then no action will occur in the windows of that model. Selecting state #n with this slider will make state #n in each window current, regardless of whether or not the times of this state in different models match.
- Contexts implying selection, e.g. **BLANK** , **WRITE** , **XY_DATA** , will force you to specify which model you plan to select from. The tabs in that panel which apply to other models will be deselected, and any attempt to select them will fail with "mixed model" error messages. Therefore you cannot pick or select from multiple models at the same time.

How models with dissimilar states are animated

By default models are animated in numerical state sequence, with no attempt being made to synchronise models or windows by time. A summary of behaviour is:

- **Multiple models in separate windows**
  Each window animates in step from frames 1 to <n>, but no window will "loop back" to frame 1 until the window with the greatest number of frames has finished. Therefore those windows with fewer frames will wait at the end of each cycle for the window with longest sequence to complete. This "stepping together" by frame does ***not*** take into account the clock time of each frame, so windows that are "in step" by frame number will not necessarily be synchronised in time.

- **Multiple models in the same window:**
  Each model in the window starts animating at state #1, and continues until the last state in that model is encountered. If one model has fewer states than another one then it remains at its last state until the other model reaches its last state. Then all models start in synchronisation at state #1 again. Again, synchronisation is "by frame" not "by time", so frames in different models may not have the same analysis time.

- **Synchronising models in time.**
  It is possible to interpolate between states, and by stipulating a fixed time interval you can synchronise animation of multiple models in time. This can be done both for models within a window, and for models across multiple windows.

This topic is covered in more detail under the **SET_STATES** command, which describes how to select what is to be animated.

5.5.10. Comparing Results Between Models

Comparing results between models

It is possible to derive results in one model with respect to another - essentially by subtracting data in Model B from that in Model A. This is done in the **DEFORM, REFERENCE STATE/MODEL** panel. Briefly:

- You can plot data and coordinates relative to a state in the current model.
- You can also plot relative to a state (or the current state) in a different "reference" model.

For example you can plot the difference between two analyses where you have changed a section property, or remeshed an area.

Data is compared using labels, for example the data for node 100 in Model B is subtracted from that for node 100 in Model A. Therefore the models need ***not*** be identical, but they do need to be topologically similar for this to work. In particular comparisons in regions which have been remeshed are likely to be unsatisfactory.

In future releases we hope to perform "geographically" based comparison, where the node or element in Model B nearest to that in Model A is used, removing the dependency upon identical labels.

5.5.11. Some Special Multiple Window Cases

Some special multiple window cases

The IMAGES and MOVIES panel

Images may be made from one or more windows, according to which tabs are selected, see Images for more information.

5.6. Checkpoint Files

Checkpoint Files

From D3PLOT 8.3 onwards, D3PLOT automatically records every command and mouse action in a "checkpoint" file. If the session terminates normally this is deleted, but if a crash occurs this file will be left on disk giving you the opportunity to recover your work.

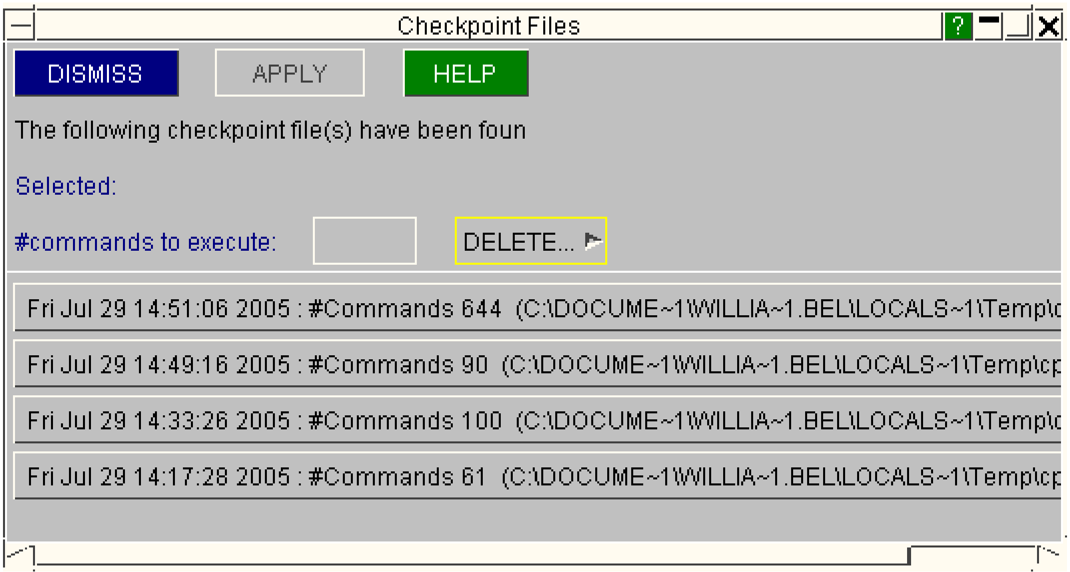
Checkpoint files have the name " **`cp_D3PLOT_21_`** *<pid>* " where *<pid>* is the current process id, ensuring that the filename is unique. They are normally written in the current directory, but if this is read-only they are written in $HOME or, failing that, in $TMP.

Please Note:
The read/write of the checkpoint files can be enabled using the " **write_checkpoint_files** " and " **show_checkpoint_files** " command-line options and/or preferences.
Also, the folder path to write the checkpoint files can be set using the " **checkpoint_dir** " command-line option and/or preference.

5.6.1. Selecting a Checkpoint File

Selecting a checkpoint file

When D3PLOT is restarted after a crash, it automatically detects any checkpoint files in the current directory (or, if this is read-only, in $HOME or $TMP). These are displayed at startup in the special **Checkpoint Files** panel:



All checkpoint files found are listed in date/time order, with the most recent file at the top of the list. To use a checkpoint file:

- Select the file to run by clicking on its row.
- Optionally delete some or all files using the **DELETE >** option.
- Optionally reduce the " **#commands to execute** " to a smaller value, perhaps to omit the last command(s) which caused a crash.
- Click on **APPLY** to run the file.

In this example the user has selected the oldest file and is about to delete the remainder using the **DELETE > ALL_BUT_SELECTED** option.

If you want to ignore all checkpoint files and run interactively just **DISMISS** this panel.

5.6.2. What Happens When a Checkpoint File Runs

What happens when a checkpoint file runs

Each command in the file is repeated verbatim, as if you had typed or mouse clicked it, until either the end of the file or the "#commands to execute" value are reached. Thereafter the session returns to being interactive in the normal way. A new checkpoint file is written which, initially, will be a copy of the one being played back, but will then contain any further interactive commands.

5.6.3. Limitations of Checkpoint Files

Limitations of checkpoint files

Although they are a powerful tool for recovering from crashes checkpoint files are not perfect. In particular they do not include any information about elapsed time between commands, which can lead to differences during playback in two situations:

- When D3PLOT animates each frame is displayed in the "dead" time between user commands. In effect the code says "Has the user given a command? No? Good, let's animate another frame while they are thinking." Animation will not actually commence during checkpoint file playback, even if **PLAY >** has been recorded, as there is no "dead" time between successive commands in which to execute it.

  Therefore if the session included animation the image which is on the screen may be different to that when the checkpoint file was recorded, and this may affect the outcome of any screen picking operations. Some more subtle consequences may also arise: for example contour bands may be different because the code has not yet decided to autoscale bands over all frames in an animation sequence.

- When the T/HIS <=> D3PLOT link is used this too may not play back correctly. The reason is that the two codes run independently and talk to one another via inter-process communication. Because checkpoint file playback leaves no intervals between successive commands, the remote programme (T/HIS) may not have had time to perform the operations requested, and return results, so the sequence of stored commands may "run ahead" of what is actually happening on the display and effectively give answers to questions that have yet to be asked.

  Therefore checkpoint file playback of all but the simplest "linked" sessions is likely to fail because of the asynchronous way in which the two codes are running.

We hope to address the issue of asynchronous behaviour in future releases, but for the time being these limitations apply.

5.6.4. Sending Checkpoint Files Back to Oasys Ltd for Debugging

Sending checkpoint files back to Oasys Ltd for debugging

We hope that you won't experience crashes but, if you do, checkpoint files can help us to find and fix the problem as they answer the question "can you tell us what you did to make it crash?".

However environment variables, settings in your oa_pref file, and any "settings" files can all influence how D3PLOT runs; and we need these to replicate the status of the code when it crashed. So when you send us crash information please could you include as much of the following as possible:

- The checkpoint file(s) themselves.
- A copy of all your " **`oa_pref`** " files: in $OASYS, $HOME and $cwd. (See Appendix B for details)
- Any "settings" files (**`d3plot`** ***`nnn`*** **`.set`**)
- Any environment variables that have been set (eg **`MENU_AUTO_CONFIRM`** , see The Following Environment Variables may be Used to Control the Behaviour of D3PLOT for a list of these)

5.6.5. Preventing the Reading and Writing of Checkpoint Files

Preventing the reading and writing of checkpoint files

When performing batch (non-interactive) post-processing, for example driven by FAST-TCF or PRESENTER, you are relying on running scripts of commands that assume a given programme state. These can be upset if checkpoint files exist on disk since they will have no mechanism for dealing with them.

Therefore the environment variable **SUPPRESS_CHECKPOINT** may be set, with the following result:

- Any existing checkpoint files will be ignored, and no questions asked about them
- No checkpoint file will be written during this session.

This and other environment variables are explained in The Following Environment Variables may be Used to Control the Behaviour of D3PLOT .

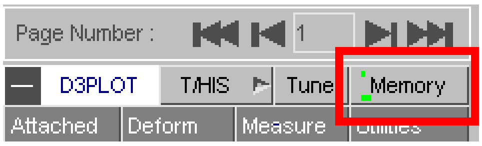
5.7. Memory Management

Memory Management

Do you need to worry about this? Unless your machine is showing signs of running out of memory the answer is no.

There is a simple way to tell if this is the case:

If the **Memory** button bars are both green you have no problems. If either turn dark orange you may need to take some action.



The top one shows this D3PLOT process size as a proportion of available physical memory on the machine. If this exceeds about 85% (and goes dark orange) the performance of the code may start to degrade as it starts to page (use virtual memory, or "swap"), although it will continue to run.

The lower one shows swap space usage (by all applications) as a proportion of total swap space available on the machine. If this exceeds about 90% you may have to take action to free some space from elsewhere. (A machine with no free swap space will simply stop, and may need rebooting!)

There are alarms built into the code which will warn you if you are approaching either of these limits, so you don't have to keep checking memory consumption.

If a memory alarm pops up ...

- The memory use of this process can be managed by controlling how much data is stored in the results database, and also by changing the animation display method.

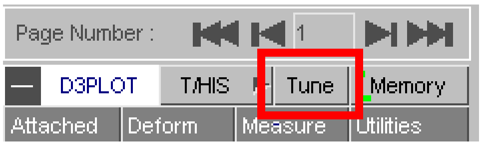
- Swap space is used by all processes on your computer, and it may be possible to free space by shutting down other processes.

A more detailed description of memory management, and the functions available under the description of the **Memory** button.

5.8. Tune Panel

5.8.1. Improving Graphics Performance

Improving Graphics Performance



Any computer intended for CAE work will have a graphics card that is designed for high performance 3d graphics. Briefly this means that:

- Most intensive graphics calculations are now performed on the Graphics Processing Unit (GPU) which has dedicated hardware for this purpose.
- Increasingly the data used to generate images can also be stored on the graphics card, since this makes access to it faster.

D3PLOT exploits these capabilities, however not all hardware is the same, and there are sometimes multiple ways of performing an operation with no obvious "best" solution. The default graphics tuning options are set up for the typical CAE machine, but if your machine is not performing well then it may be worth trying some of the suggestions below in order to try to improve matters.

The default settings are tuned for the typical machine

When you first use the **Tune** button you will see this panel with all boxes unticked.

This means that none of the special hardware accelerations available in V10.0 are switched on, and graphics performance will be similar to that in V9.4. You will have to tune D3PLOT manually to obtain the best performance.

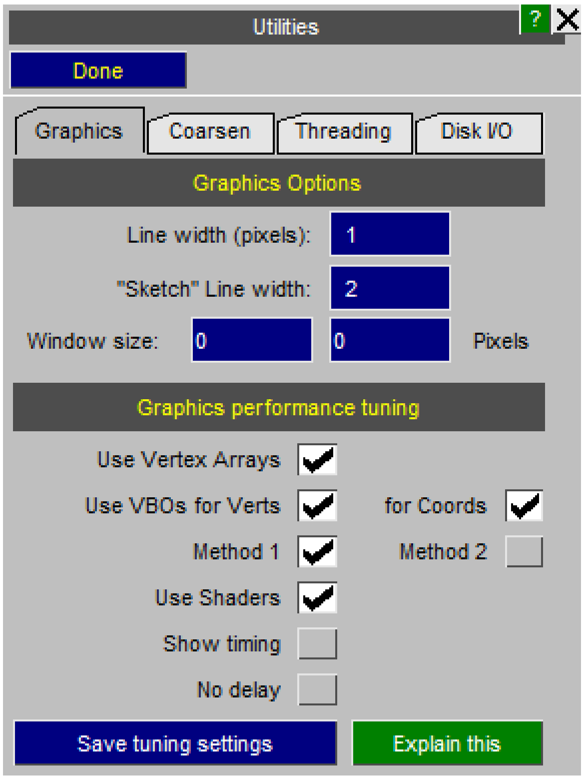
If some or all buttons are greyed out it means that your hardware does not support the feature in question, and you will not be able to use them. Therefore re-tuning is always advised if you move to a new machine, or upgrade your hardware; and it may also be worthwhile if you have upgraded your graphics driver software.

Why not "tune" automatically?

Given that D3PLOT can determine automatically the features available on the graphics card and supported by the graphics drivers why can it not also set these options automatically?

The unfortunate answer is that what a graphics hardware/software combination ***says*** it will do, and what it will ***actually*** do, are not always the same. Bitter experience has taught us that on some machines, typically slighter older ones and/or those with lower performance, blithely turning on all possible acceleration features can lead to corrupted images and/or crashes.

Therefore we prefer to default to something that we know will work on all machines, and let users determine what works best on their hardware using the process below.



Tuning D3PLOT on your machine

Assuming that your machine doesn't have all the tuning buttons greyed out, which means that it is too old to benefit, please follow the steps below to tune its performance.

Step 1: Read in a sizeable model with some state data so that you can animate it. You just need one model in one window.

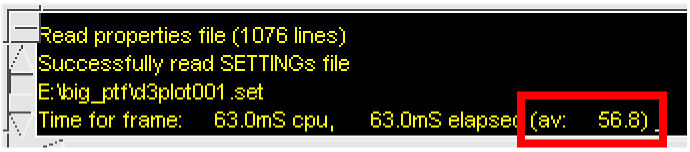
The model should be large enough to give animation rates well below the maximum your machine can support. Since the typical modern LCD display is limited to a 60Hz refresh rate there is little point in aiming for animation rates faster than 60 frames per second (fps), and you should aim for a model that is giving about 50 fps or slower, that is an elapsed time of 20ms/frame or longer.

Set the model animating in shaded mode and let it cycle through at least one pass of the full animation so that it has read in all data and settled down to its full animation rate.

Standard shaded settings should be used: perspective off, a single light source, flat shading, free-edge overlay and no "extra" graphics such as labels.

Step 2: Use the **Tune** button to invoke the tuning panel, and turn on the following two options:

	**Show timing** reports the time taken for each frame in the dialogue box. Three times are given: 1. CPU time required to generate this frame 2. Elapsed (wall-clock) time required to render this frame 3. A rolling average over the last 30 frames of "elapsed time to render frame"
	**No delay** turns off the default setting in D3PLOT that limits the maximum frame rate to about 60 fps. This means that there are no artificial delays in the timing process, and the steps below will be measuring the true performance of your machine.

	Observe and note the rolling average time per frame in the dialogue box. This should be a reasonably steady figure.
Don't be surprised if the " **`Av`** " figure is slightly different to both " **`cpu`** " and " **`elapsed`** " values. Timers on computers tend to have a limited resolution, for example Windows machines run at a "clock tick" of 60Hz, and only resolve time intervals down to roughly 16mS as a consequence. This is why the rolling average frame rate is required in order to smooth out variations in individual frame timings.	

Step 3: Turn on **Use Vertex Arrays**

<table>
<tr><td>
</td><td>This will make no difference to the current animation speed, but it is a necessary precursor to the steps below.</td></tr>
</table>

Step 4: If the **Use Shaders** button is greyed out then please skip to step 5 below, otherwise:

<table>
<tr><td>
</td><td>Turn on Use Shaders.

You will hopefully see an immediate and significant reduction in the time taken to render frames, but otherwise the appearance of the image should not change. If this is the case leave this option selected and proceed to step 5.

On some machines the model may in fact animate more slowly with this setting. In this case it is worth persevering with step 5 below to see if adding the further settings does ultimately give better speed.</td></tr>
<tr><td colspan="2">If the image goes "wrong" in any way, and we have observed everything from losing colours, through a totally corrupt image to an outright crash, don't despair. The first thing to do in this situation is to try updating the machine's graphics driver. This will require you to determine the type of card on the machine, then to visit the card manufacturer's website, download the appropriate driver and install it. If you are not sure how to do this Oasys Ltd can advise you, so please contact us for help.

In about 90% of cases this will solve the problem, but if it doesn't then you will not be able to use hardware shaders and you need to turn this option off and proceed to step 5.</td></tr>
</table>

Step 5: If the **Use VBOs for Verts** and **for Coords** buttons are not available available please skip to step 6 below, otherwise:

<table>
<tr>
<td>
</td>
<td>Firstly turn on the (Use VBOs...) for Coords button, leaving the for Verts one unselected for now.

You should hopefully see a further significant increase in speed, but in all other respects the image should look as before. If this is the case then...</td>
</tr>
<tr>
<td>
</td>
<td>Turn on Use VBOs for Verts .

The effect of this varies by hardware type and can range from a small but significant increase in speed, through not much change to a slight slowing down. If the effect is neutral or positive then it is worth leaving it selected, but otherwise it is better to turn it off.</td>
</tr>
<tr>
<td colspan="2">As above, if the image goes "wrong" in any way with either of these settings then the first thing to do is to update the graphics driver. If this does not help, and only Use VBOs for Verts is causing problems, then you can leave it turned off without sacrificing much performance.</td>
</tr>
</table>

Step 6: If all the steps above were successful then you have finished the D3PLOT tuning process. Hopefully you will have achieved a significant speed increase and the final step is to save these tuning settings in your oa_pref file for future sessions.

Save Tuning Settings will do this automatically, saving the relevant entries to your "home" oa_pref file.

If you want to copy these settings to the same file for other users the preferences in question are:

```
d3plot*gtune_varray
d3plot*gtune_shader
d3plot*gtune_vbo_verts
d3plot*gtune_vbo_coords
```

If things went wrong above, or some options are not available on your machine, then you may still benefit from using the settings that are available and seem to work. If you need further advice please contact Oasys Ltd. for help.

Updating and configuring your graphics driver

This section gives instructions for updating graphics cards on Windows and Linux machines. It covers the most common configurations, but if your machine is different then it is possible that we may still be able to help you, please contact us for advice.

The vast majority of graphics-related support requests that we receive are solved by upgrading the graphics driver.

Our general advice is as follows:

- Assuming that your graphics card is reasonably modern (typically a machine less than 5 years old) you should check that you have the most recent graphics driver, and update it if you haven't.

  Manufacturer's websites all have pages that allow you to enter details of your machine and operating system, and they will then recommend the best driver.

  For NVidia cards see http://www.nvidia.com and use the "drivers" tab
  For AMD (ATI) cards see https://www.amd.com/en/support
  For Intel cards see
  https://downloadcenter.intel.com/product/80939/Graphics-Drivers

  You can also visit the computer manufacturer's website to find the driver they recommend. However it is our experience that the computer manufacturer's recommendation is usually out of date, and that you are better off using what the graphics card manufacturer recommends. They made the hardware, they write the drivers and they generally know best!

- For an older machine, say 5 or more years old, you will probably find that the manufacturer will recommend their latest driver, but this may not perform as well with an older card. Graphics drivers tend to be optimised for recent hardware, and old hardware may actually run more slowly - albeit reliably - on newer drivers.

  In this situation it is worth investigating when your graphics card first appeared, then visiting the manuafacturer's driver archive and trying to find a version that is about 1 to 2 years newer than the card. This should have the bugs ironed out, but should not have suffered too much from changes aimed at newer hardware that degrade its performance.

- It is almost universally the case that the "out of the box" graphics driver which comes with a new machine will be out of date. If your machine comes with a very recent latest and greatest graphics card then the driver will be an early one and will probably be bug-ridden. By the time the machine reaches your desk things should have moved on a bit, and it is worth checking for a newer version - it can save a lot of grief.

This is usually also the case with corporate builds of Windows where your organisation gets a customised version of the Windows operating system that is blindly loaded onto all machines by the computer manufacturer. Such builds are usually even more out of date than the "out of the box" ones above and quite often contain poorly performing generic graphics drivers rather than the specific ones that high performance GPUs require.

Finding out what graphics card and driver you have installed

The following instructions should enable you to determine the type of graphics card you have installed and the revision number of its driver.

Windows 10

- Right click anywhere on the desktop background, and select **`Display Settings`**
- Near the bottom of that panel select **`Advanced display settings`** . This will give you a summary.
- Select **`Display Adapter properties`**
- Select **`Properties`** within this section, followed by the **`Driver`** tab
- This will list the driver date and version

  Alternatively:

- Launch the Control Panel and search for " **`Device Manager`** "
- Click on that, expand the tree and then expand the **`Display Adapters`** branch
- Right click on your graphics card's row and select **`Properties`**
- Select the **`Driver`** tab. This will list the driver date and version

  Alternatively:

- Your graphics card manufacturer may well have an icon in the icons region at the bottom right of the taskbar, or a right click anywhere on the desktop background may show a vendor-specific entry in the popup menu.

Windows 7

- Right click anywhere on the desktop background, and select **`Screen Resolution`**
- Select **`Advanced settings`**
- This takes you to the **`Adapter`** window, listing card name and manufacturer
- Select **`Properties`** within this section, followed by the **`Driver`** tab

- This will list the driver date and version

Linux

- type **`glxinfo | grep -i string`** which should give the card manufacturer and name

For example on a machine with an ATI card this produces:

```
OpenGL vendor string: ATI Technologies Inc.
OpenGL renderer string: ATI FirePro V7750
(FireGL)
OpenGL version string: 3.3.10225 Compatibility
Profile Context FireGL
```

And on a machine with an NVidia card this produces:

```
OpenGL vendor string: NVIDIA Corporation
OpenGL renderer string: Quadro FX 3800/PCI/SSE2
OpenGL version string: 3.3.0 NVIDIA 256.35
```

- Knowing the make of card you can then look in file /var/log/Xorg.0.log for more details. For example in the 2nd example above **`grep -i nvidia /var/log/Xorg.0.log | grep -i driver`** gives:

```
(II) Loading
/usr/lib64/xorg/modules/drivers/nvidia_drv.so
(II) NVIDIA dlloader X Driver 256.35 Wed Jun 16
18:45:02 PDT 2010
(II) NVIDIA Unified Driver for all Supported
NVIDIA GPUs
```

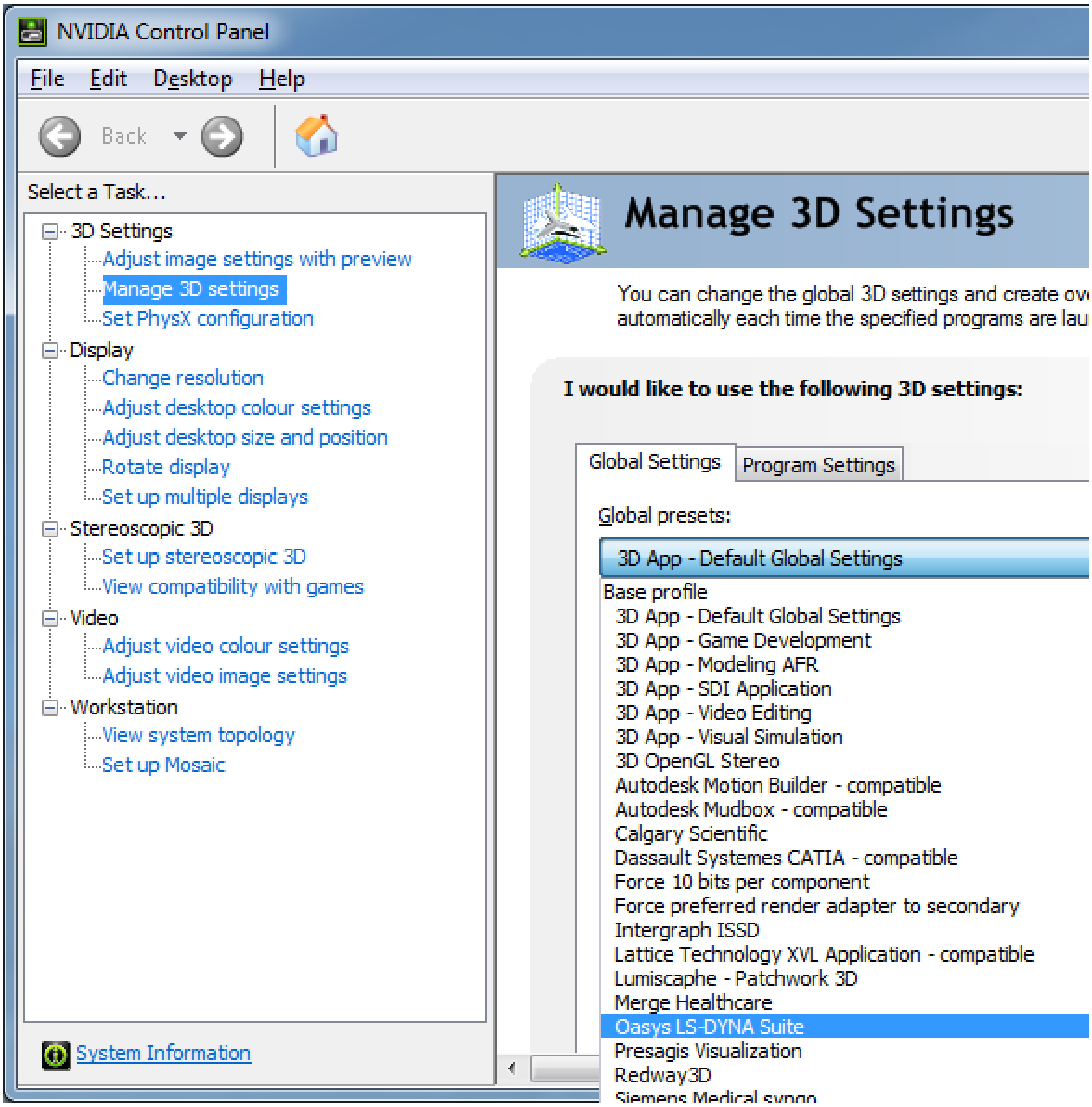
So this 2nd machine has an NVidia Quadro FX3800 card using driver release 256.35 dated June 16th 2010

Configuring NVidia cards on Windows

Recent installations

- Right click anywhere on desktop background, and select **`NVIDIA Control Panel:`**

  Select **`Manage 3D settings`** from the tree on the left hand side. The example below is from a Quadro FX card on a Windows 7 machine, but others should be very similar.



- You must then decide whether you want to configure the graphics driver for all applications on your machine or just for a limited range of executables.

  Our recommendation is to configure for all applications, using **`Global settings`** as shown above. The configuration used should work well for any CAE package - and certainly better than NVidia's default " **`3D App -Default Global Settings`** ", since these are tuned for benchmark tests and not real life applications.

  If you want to apply settings only to D3PLOT you will need to swap to the Program Settings tab, add D3PLOT to the list, and then proceed as below.

- If your driver is recent (early 2011 onwards) you will find an **`Oasys Ltd. LS-DYNA environment`** setting as shown above, and you should select that. If your driver is older we would recommend using **`Dassault Systemes CATIA - compatible`** .

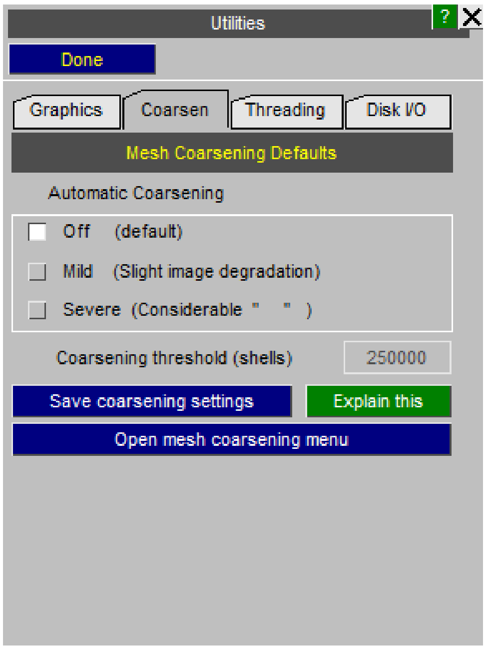
  Either of these settings turns off attempts in the driver to cache coordinate data, and will result in smooth animation. Using the default settings may lead to jerky animation, or long pauses.

Configuring NVidia cards on Linux

No configuration is necessary.

5.8.2. Mesh Coarsening

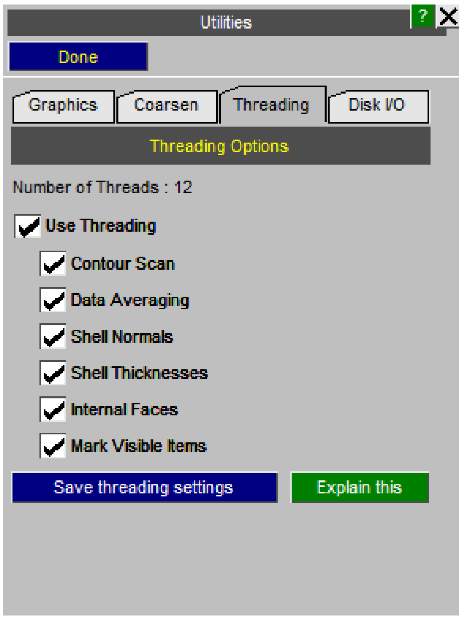
Mesh Coarsening



See UTILITIES, COARSEN for more detail.

5.8.3. Threading

Threading



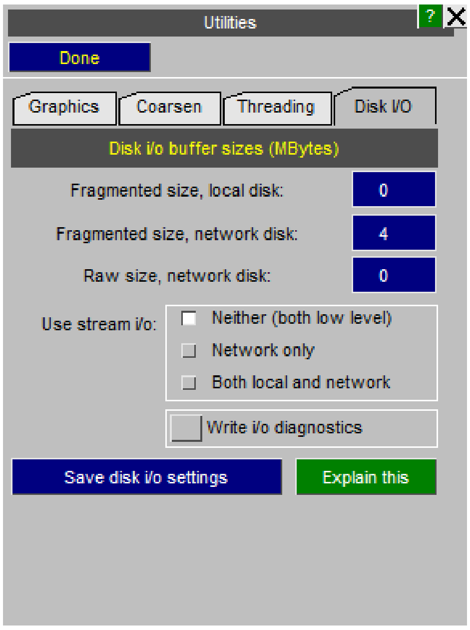
Several operations in D3PLOT can be run in parallel on multiple processors. This generally speeds them up, but it can be turned on or off here for the following operations:

- Contour Scan (finding minimal/maximal values of data components for the contour bar)
- Data Averaging at nodes (used as part of contour calculation when averaging is turned on)
- Shell normal calculations (for graphics update when hardware shaders are turned off)
- Shell thickness calculations (used to calculate surfaces when the new "thickness" option for shells is turned on)
- Internal face calculations (used when blanking items or turning on/off display of entity types)
- Marking visible items (used when blanking items or turning on/off display of entity types)

As the results do not depend on whether or not threading is active, it is recommended to leave these options on. The only reason for providing them is for the case of unexpected behaviour. In addition to turning them off for the current session here, they can also be saved as preferences.

5.8.4. Disk i/o Buffer Size Settings

Disk i/o Buffer Size Settings



D3PLOT has to read a lot of analysis data from disk, and it improves performance if this process is tuned.

It is convenient to separate disk into "local" (inside this machine) and "network" (accessed remotely), since the sizes that work best in each case vary. Therefore D3PLOT has the following size settings:

- **Fragmented local** : Used to read interleaved element data from local disk into a buffer.
  On most systems this works best if it is set to "unlimited" size, denoted here by zero bytes. On 64 bit systems this is genuinely unlimited, however on 32 bit systems this implies a cap of 100MBytes to limit the amount of memory dedicated to i/o. On all systems there is also a "hard" limit of 2 GB since values greater than this can cause problems.
  It is recommended that this value is left unchanged.
- **Fragmented network** : Used to read interleaved element data from networked disk into a buffer.
  This defaults to 4 MBytes, a value arrived at by trial and error on a range of networks. Tuning this value can have a significant effect on the time taken to read data, and you will need to experiment with your network to find out what works best for you.
- **Raw network** : Used to read sequential nodal data directly from networked disk.
  Nodal data is stored on disk in a format that permits it to be read directly without the need for an intermediate data buffer. On most systems the optimum performance is obtained (on both local and network disks) by reading it as a single chunk of data in one operation; however we have encountered networks that place an artifical limit on the amount of data that can be read in this way. It is also possible that some networks will give better performance if the amount of data read in a single operation is limited.
  Therefore the default read size is "unlimited", denoted by a size of zero bytes,

and in most cases this will give the optimum performance. Experimenting with finite sizes *may* give improved performance, but it is our experience that this value is best left as "unlimited" unless you find that you are having network problems reading nodal data.
Note that changing any of the sizes above will only apply to models opened after the change, so to see effects it will be necessary to "reopen" models.

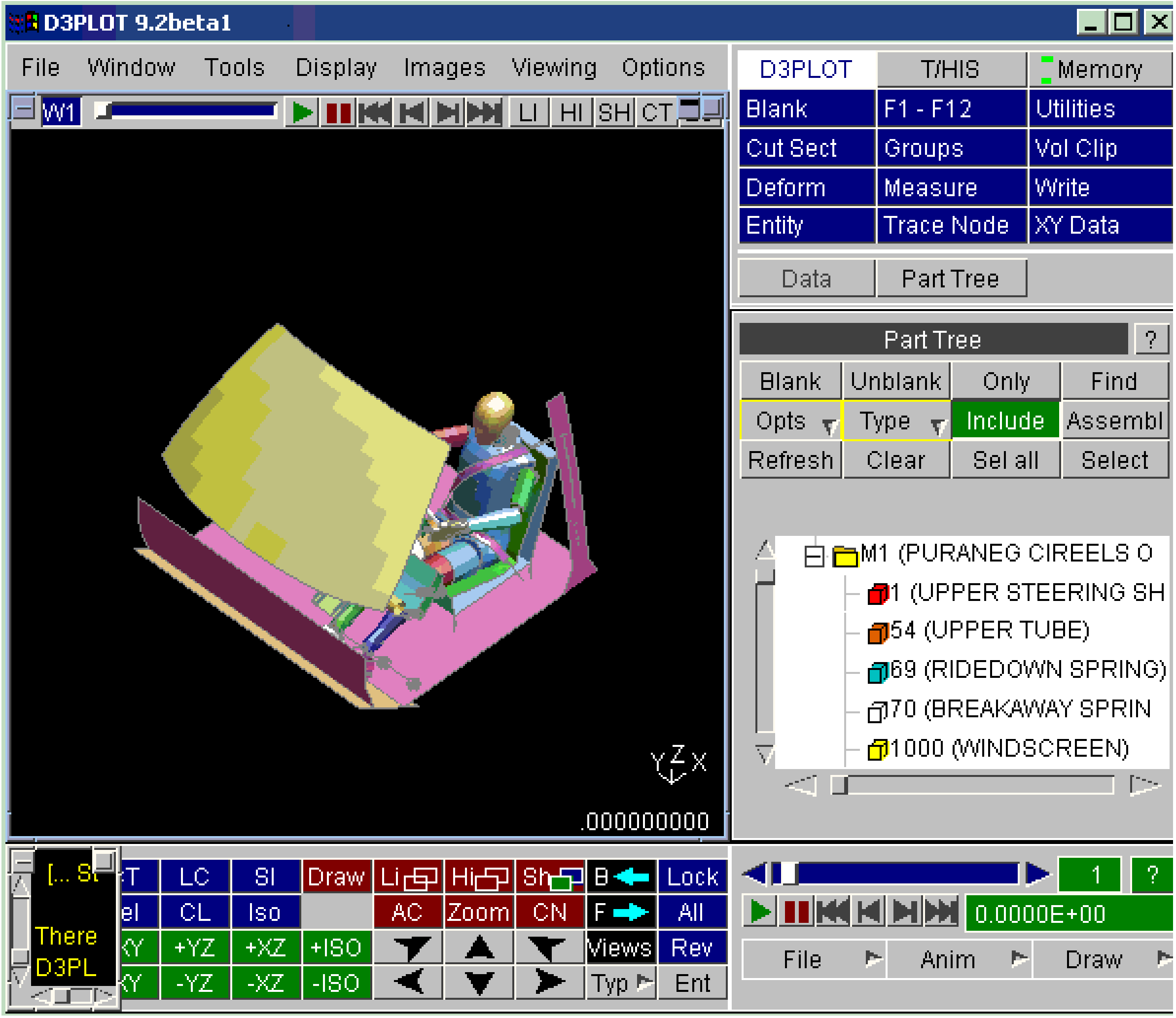
- **Using "stream i/o"** : Which system routines are used: low level raw or higher level buffered.
  Operating systems provide both direct low-level raw access to disks and higher level buffered access, referred to as "stream i/o". Generally the unbuffered low-level method is faster, especially for local disks, but on some networks it is sometimes possible to obtain better performance with buffered i/o.
  If you have network speed problems it may be worth trying buffered stream i/o to see if you obtain better performance but this is something you will have to determine by trial and error as every network is different.
  (If you choose stream i/o then the buffer sizes set above are ignored because they are irrelevant.)
  Changes to any of the above only take effect when a file is opened. To apply them to a currently open set of files it will be necessary to close and reopen the model. This can be done via File -> Reread model. Buffer sizes and stream i/o settings can be saved in your oa_pref file using the "Save disk i/o settings" button.
- **Write i/o diagnostics to log file** : This logs all disk i/o operations to file and is used for diagnosing i/o problems, Oasys Ltd may ask you to turn it on and to send us the resulting file. It can also be set by creating environment variable **`D3PLOT_DIAG_OUTPUT`** to **`TRUE`** .

6. Using the D3PLOT Screen Menu System

6.1. Basic Screen Menu Layout

Basic screen menu layout

D3PLOT runs within a single window, owned by the window manager, which has several sub-windows inside it. A typical D3PLOT session will look like this:



The various sub-windows always exist within the master window, and may be moved and resized at will inside it. They will keep their relative size and position as the master window is changed in size and/or shape, and will reappear after the main window is de-iconised.

Their exact location and size will depend on the size and resolution of the display: you can use the **DISPLAY_FACTOR** variable (see [If D3PLOT will not open a window on your display](#)) to override default sizes and resolution.

The **TIDY** button in the icons box may be used at any time to restore this default layout: any unwanted sub-windows will be closed and the screen will be restored to the appearance here.

The default layout of the main sub-windows is as follows:

<table><tr><td colspan="2"></td><td rowspan="2">Main Menu Options</td></tr><tr><td colspan="2" rowspan="2">Graphics Area</td></tr><tr><td>Menu Area</td></tr><tr><td>Dialogue and List</td><td>Global Commands</td><td>Animation Controls</td></tr></table>

Main Menu Options	Provides access to the majority of the commands and options available in D3PLOT through a series of sub menus
Graphics area	Is where graphics are drawn.
Dialogue & list	Allows "command-line" input and output, also provides a listing area for messages.
Menu Area	Displays the commands and options associated the current selection from the main menu options.
Global Commands	Gives access to commonly used commands
Animation Controls	Controls states and what is displayed during animation

While you are free to re-position these master windows it is recommended that you keep to this default layout. This is because when further sub-windows appear their position and size is designed assuming this layout, and aims to obscure as little useful information as possible.

6.2. Mouse and Keyboard Usage for Screen Menu Interface

Mouse and keyboard usage for screen-menu interface

All screen-menu operations are driven with the left mouse button, with the following exceptions:

Text in the dialogue area and text boxes requires keyboard entry;

Text strings saved in the cursor "cut" buffer may be "pasted" into dialogue areas and text boxes using the middle mouse button.

Popup" menus are invoked using the right mouse button.

The primitive "widgets" in the menu interface are used as follows:

BUTTONS:

Screen buttons are depressed by clicking on them, but action only takes place when the mouse button is released, so it is safe to drag the (depressed) mouse around the screen.

Buttons may be set (i.e. depressed) by D3PLOT itself, for example the **Solids & Shells** one above, to indicate that this option is in force. They may also be greyed out, for example the **Cont Surfs** one above, to indicate that the option is not currently available. Some buttons repeat automatically when held depressed: this depends on context. Buttons with " **...** " after them will invoke sub-menus.

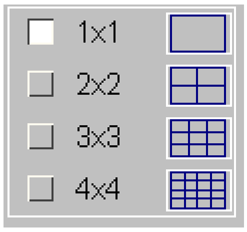
SLIDERS:

Sliders are moved by clicking on the slider button itself, and then dragging it to a new position. They may also be moved automatically by clicking on, and holding down, one of the arrows at either end.

TEXT BOXES:

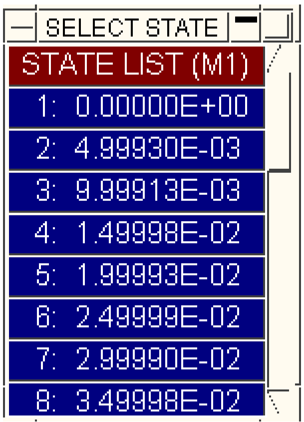
To enter text in a text box: first make it "live" by clicking on it, then type in text, then type **<return>** to enter the string. Clicking on a "live" box for a second time is exactly the same as typing **<return>** , so clicking twice on a box effectively enters its current contents. You can use the left and right arrow keys for line editing within a box: text entry takes place after the current cursor position.

RADIO BOXES:



A "radio" set is provided where only one selection is possible from a range of options. In this example the laser postscript output has been set to a single image per page. To select click anywhere on the row of the relevant option, any previously selected item will be deselected.

MENU LISTS:



Menus of items are used when you need to make one or more selections from a (potentially) long list. Click on the row you want to select: clicking on a row that is already selected will have the effect of unselecting it. When the list is too long to display in the window you can use the vertical scroll-bars to move up and down it.

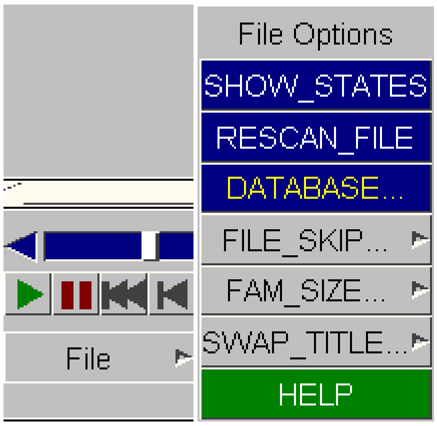
POPUP MENUS:

Where a button has a "right arrow" **>** symbol it means that a popup menu is available.

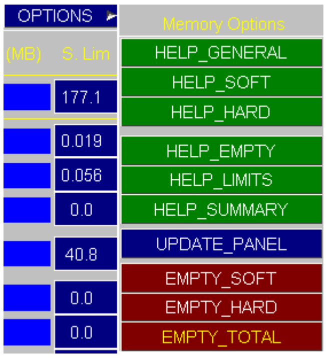
Click the right mouse button and the menu will appear. Holding down the right mouse button drag it onto the item you want.

Popup menus can be nested to any depth.

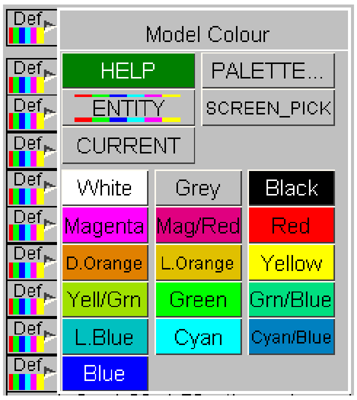
Note that popup menus can be invoked both from "clickable" buttons and from "non-clickable" ones: it makes no difference to their functionality.



Popup menu invoked from "clickable" **OPTIONS>** button



Popup menu invoked from "non-clickable" **DEF>** (colours) button

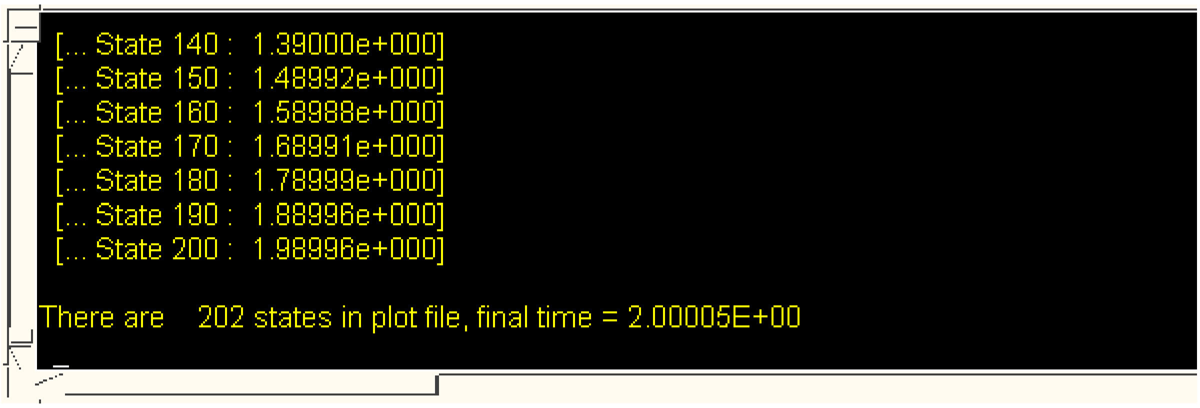


6.3. Dialogue Input in the Screen-Menu Interface

Dialogue input in the screen menu interface

The full command-line capability is preserved when D3PLOT is running in screen-menu mode, and you are free to mix command-line and mouse-driven input at will. There are some situations in which command-line input is more efficient: for example when entering lists of explicit entities.

Commands are entered in the dialogue box:



As this example shows the dialogue box is also used for listing messages, warnings and errors to the screen. It can be scrolled back and forth (its buffer is 200 lines long) to review earlier messages. The following colours are used:

Normal messages and prompts	Yellow
Text typed in by you	White
Warning messages	Magenta
Error messages	Red

There is a minor limitation when mixing command-line and screen-menu mode: you cannot perform the same function simultaneously in both modes. If you attempt to do so you will get the message:

```
WARNING: recursive access attempted
```

And you will not be permitted to continue. To clear this situation either close down the menu-based operation, or return to the main menu (" `/` " command) in the dialogue box.

6.4. Window Management in the Screen Interface

6.4.1. Basic Operations

Basic operations

Moving, resizing and scrolling of windows is based on the conventions used in the Motif Window Manager.

To move a window:	Click down on its title bar, then drag the window to where you want it to be. A "rubber-band" outline moves to show the window's current position. Where a window does not have a top title bar click anywhere on its grey background and drag it.	
To resize a window:	Either	Click on a border bar to move just that side, or on a corner bar to move both sides attached to that corner. Again, a rubber-band outline shows you the new shape.
	or	Use the **MAXIMISE** button in the top right hand corner of the window to increase the size of the window to the largest required size.
To scroll a window:	If a window has got too small for its contents then horizontal and/or vertical scrollbars will appear. Click on a scrollbar slider and move it to the desired position, the window contents will	

scroll as you do so. Alternatively click on the arrows at either end of the scrollbar for timed motion in that direction.

To iconise a window:

Click on the **ICONISE** button in the top right hand corner of the window.

To restore a window:

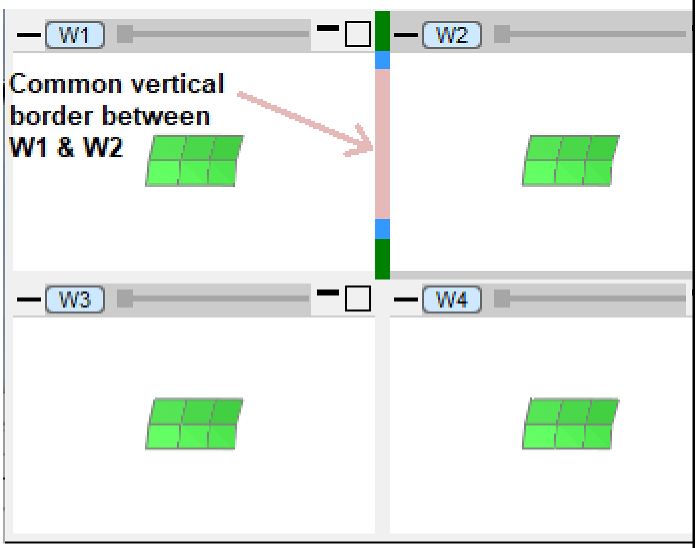
Iconised windows may be restored by clicking on the icon in the **ICON** area.

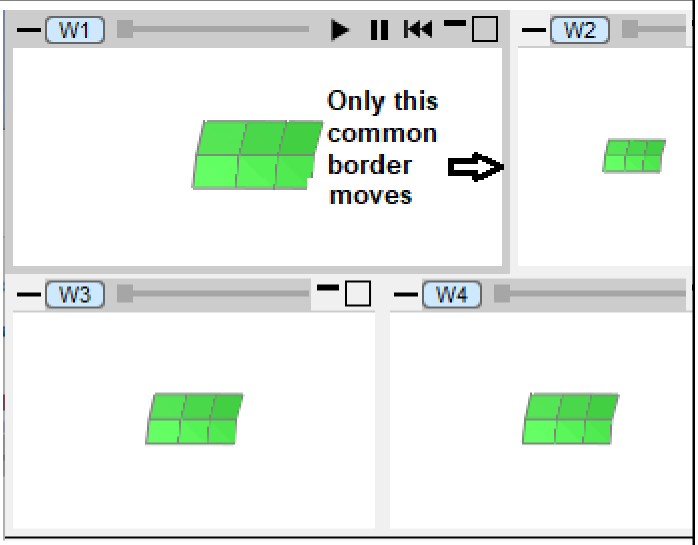
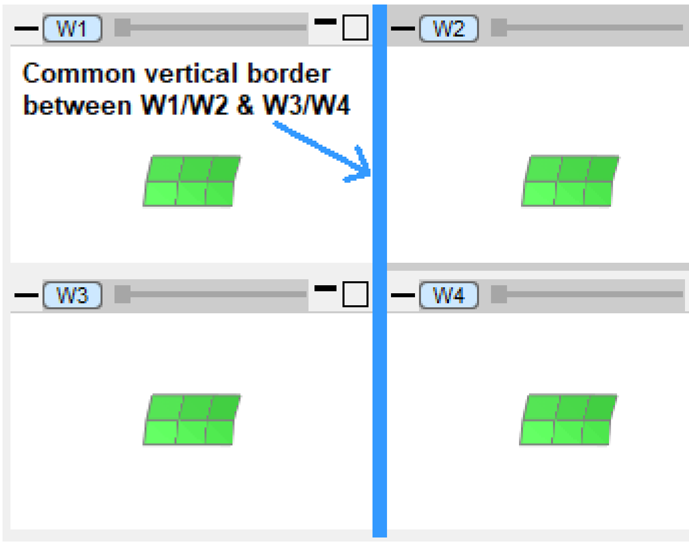
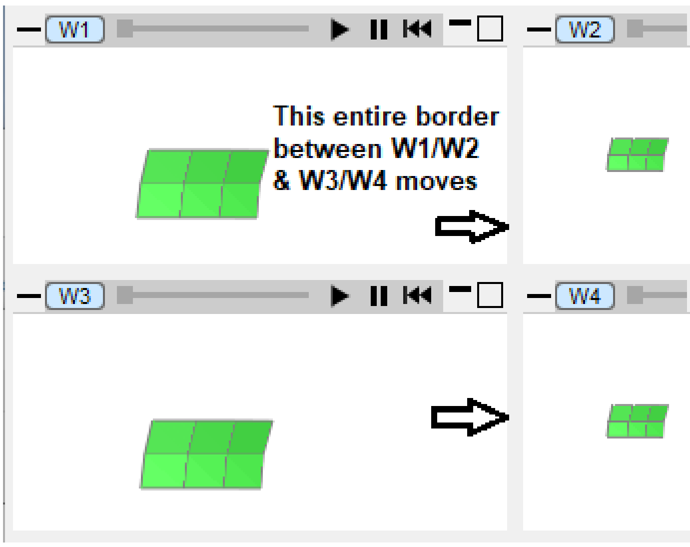
6.4.2. Common Borders for Graphics Windows

Common Borders for graphics windows

When a page contains more than one graphics window, these are laid out in a cellular grid as defined in the Window Layout section. This leads to "common borders" between adjacent windows. From D3PLOT 19.0 onwards, it is possible to drag common borders with the mouse in a way that resizes windows on both sides of the border as shown in the following images.

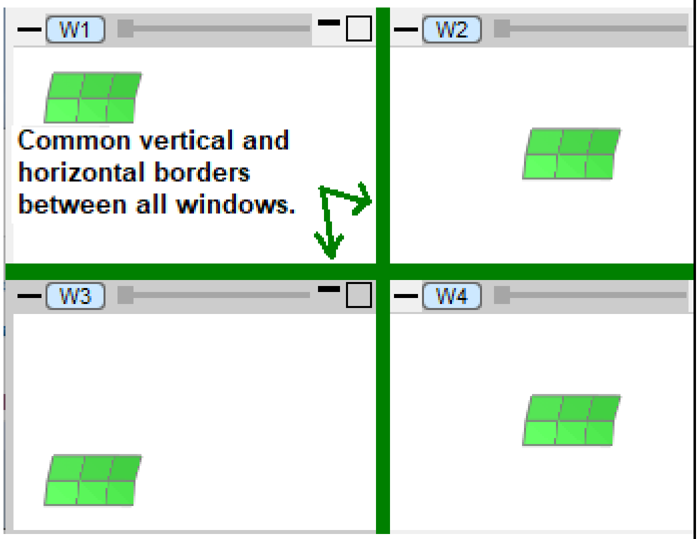
Move the mouse (don't depress a button) over a border region between two windows. This will highlight the drag areas in which a "click and drag" operation will move borders. In order to control which borders are dragged, three zones coloured pink, blue and green are shown and these have the following meanings:

Pink zone defines a common border between exactly two adjacent windows.  (Horizontal and vertical borders behave the same way. A horizontal border is shown here. The example on the right shows a vertical border being moved.)	

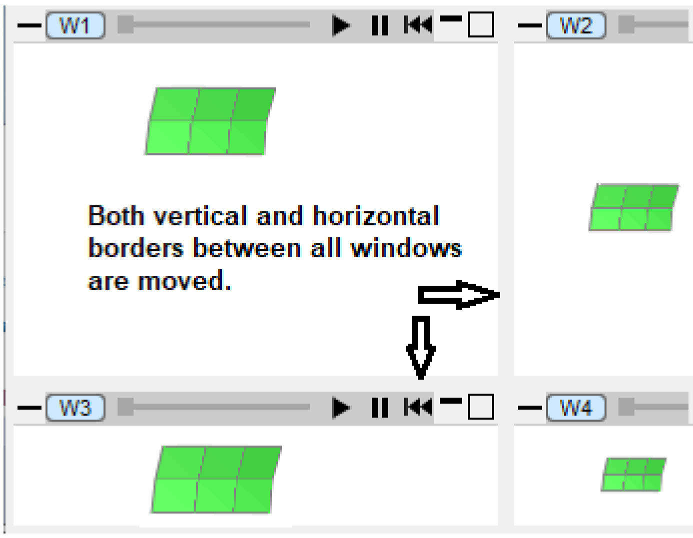
Dragging in this pink region border moves only that common border between the two windows. So in this example the vertical border between W1 and W2 is moved, but that between W3 and W4 is unchanged.	
Blue zones define a common border extending the full height or width of the page as appropriate.  (Both blue zones have the same effect, it doesn't matter which end you use.)	
Dragging in the blue zone moves all windows on either side of the border in the appropriate direction. In this example, all four windows are moved.	

Green zones define two common borders extending both horizontally and vertically to the full width and height of the page.

(Both green zones have the same effect, it
doesn't matter which end you use.)

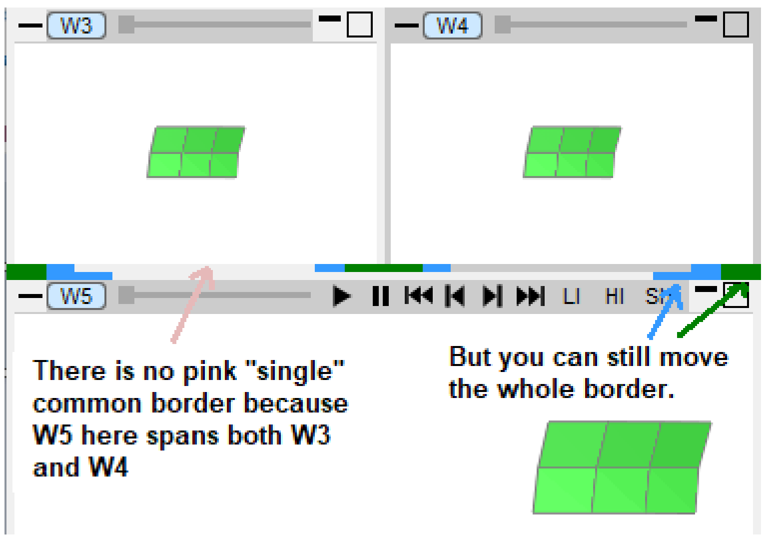


Dragging in the green zone moves all windows on either side of the border in the appropriate direction. In this example, all four windows are moved.



When windows are not the same size:

In this example, W5 is twice the width of W3 and W4 above it so there is no single common border between W3/W5 or W4/W5. In this situation there will be no pink zone, only blue and green.



Positioning the mouse at window edges:

When using the green zone to drag both horizontal and vertical axes, the borders that are dragged are those which intersect at the corner where the mouse is located.

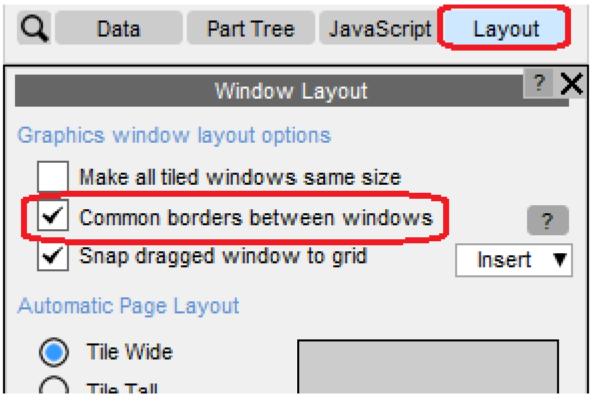
In this example, the mouse is at the bottom left of W1 / top left of W3 and it can be seen that the borders which are highlighted for dragging are those which intersect at this point.

Switching common border dragging on/off

Common borders are on by default, but they can be controlled from the [Layout]() panel.

The default behaviour may also be set by the preference:

```
d3plot*common_window_borders: true | false
```



6.4.3. Window "Snap to Grid" and Other Options

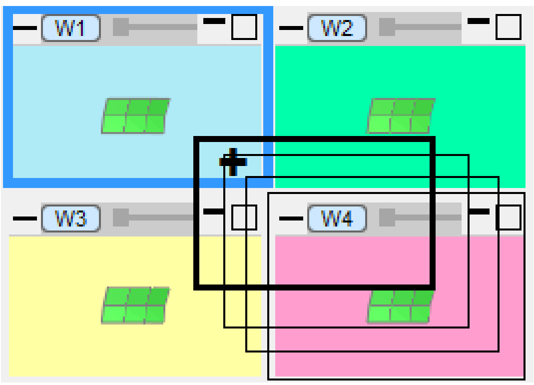
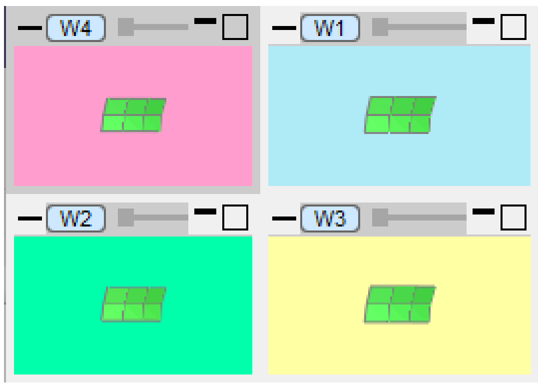
Window "Snap to Grid" and other options

When dragging an entire window with the mouse to move its position on the screen (i.e. not resizing it) there are several possible outcomes:

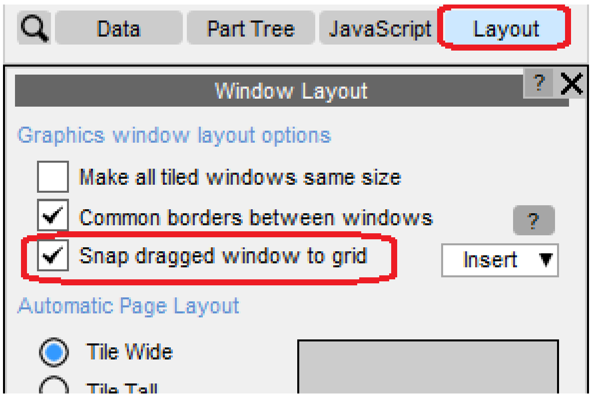
1. **Snap to Grid** : The window is moved from one "cell" in a multi-window page to a different cell, shifting the contents of one or more cells out of the way.

2. **Free positioning #1** : The window is moved from inside the D3PLOT master window to a new user-defined position within that window, i.e. positioned where it is "dropped".
3. **Free positioning #2** : The window is moved from inside the D3PLOT master window and out onto the desktop.

The behaviour of "Snap to Grid" is illustrated in the following figure:

<table>
<tr>
<td>In this example, W4 is being dragged and the cursor is
at the "+" position inside W1. The borders of W1 are
highlighted in blue to show it is the destination window.

</td>
<td>Releasing the mouse in this example causes W4 to
move to the "cell" of W1, and windows W1 to W3
to move down in the grid to make space for it.

</td>
</tr>
<tr>
<td colspan="2">"Snap to grid" behaviour illustrated in "Insert" mode</td>
</tr>
</table>

Switching "Snap to grid" on/off



Snap to grid is on by default, but it can be controlled from the **Layout** panel.

The default behaviour may also be set by the preference:

`d3plot*snap_window_position: true | false`

If switched off, the window positioning within the master D3PLOT window reverts to "Free positioning #1" mode with the window positioned where it is dropped with the mouse.

The behaviour of the other windows when a window is moved into a new position depends on whether the mode is **Insert** or **Swap** :

Insert	Other windows circulate either up or down, as in the example above
Swap	The window being dragged and its destination window swap places

Dragging a window from inside the D3PLOT master window onto the desktop, "Free positioning #2", is independent of the "snap to grid" setting: once on the desktop the window does not have any cell membership.

The ordering of windows within cells can also be controlled explicitly within the the **Layout** panel.

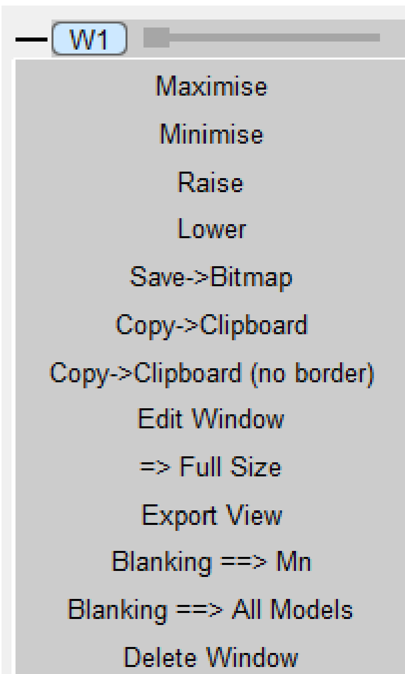
6.4.4. Further Options

Further options:

Click on the **OPTIONS** button to get the drop-down menu of window management options:

Restore	Restores a maximised window to its previous size, only shown if window is maximised
Expand	Expands an iconised window, only shown if window is iconised
Minimise	Iconises this window (see below for iconisation)
Maximise	Maximises this window to fill the whole area
Raise	Raises this window to the top of the stacking order.
Lower	Lowers this window to the bottom of the stacking order.

Save->Bitmap	Generates a windows bitmap (`.bmp`) file of the sub-window. This is an uncompressed file with a depth matching the number of bit-planes of the window. (This often doesn't work for the graphics window, since it uses mixed "visuals": use the **IMAGE** option instead for this.) For text (ie dialogue or listing) windows places a copy of the complete window text onto the system clipboard.
Copy->Clipboard	***On Windows platforms only*** : for other window types places a bitmap of the window into the clipboard as an image. (This option is not available for non-dialogue windows on X11-based window managers under Linux and Unix.)
Copy->Clipboard (no border)	Same as Copy->Clipboard but without the window border.
Edit Window	Launch the Edit Window menu (see EDIT WINDOW Changing "Model in Window" attributes).
->Full Size	Make the window the full size of the graphics area.
Export View	Export the view of the window to all other windows.
Blanking ==> Mn	Export the blanking of model in window to same model in all active windows.
Blanking ==> All Models	Export the blanking of model in window to all models in all active windows.
Delete Window	Delete the window.



6.5. "QUICK PICK" Options

"QUICK PICK" Options

Across the top of the graphics area is a block of 2 buttons. These are referred to as the **QUICK PICK ENTITY** button and the **QUICK PICK OPERATION** button in the following section.

By default the **ENTITY** and **OPERATION** buttons are set to **PART** and **BLANK.**

Possible operation types are:

BLANK (default)	BLANK "Blanking" Controls the Visibility of Nodes and Elements	
UNBLANK	BLANK "Blanking" Controls the Visibility of Nodes and Elements	
ONLY	BLANK "Blanking" Controls the Visibility of Nodes and Elements	
EDIT	Quick Pick Edit	}These options are }only available when }linked with PRIMER
PART TABLE	Quick Pick Edit	
INFORMATION		
COLOUR	Properties: Controlling colour, drawing style, transparency, lighting attributes and overlay of entities	
TRANSPARENCY	Object Attributes	
DISPLAY MODE	Display Modes	
LABEL	Labelling	

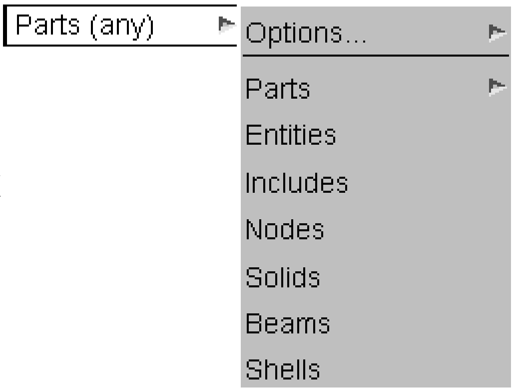
OVERLAY MODE	OVERLAY... Controlling the Hidden-Line Overlay of Element Borders On Data Plots
OVERLAY COLOUR	OVERLAY... Controlling the Hidden-Line Overlay of Element Borders On Data Plots
BRIGHTNESS	The "material" properties of the lit objects
SHININESS	The "material" properties of the lit objects
SHELL THICKNESS	Drawing Shells with Thickness
LOCATE IN TREE	Part Tree
TRACE	Trace
TARGET MARKER	TARGET_MARKERS Adding "Target" Symbols on Nodes
XY DATA (Only available by right-clicking on an entity)	XY_DATA Drawing Numerical Data as XY Plots and/or Writing it to File
PROPERTIES	Properties: Controlling Colour, Drawing Style, Transparency, Lighting Attributes and Overlay of Entities

Whenever these buttons are visible in a graphics window "quick picking" is active, and the cursor is live. Mouse buttons have the following functions:

`Left`	"Do" the operation. For example blank the entity if in BLANK mode, change its colour if in COLOUR mode, etc. Drag across the screeen using the left mouse button to selecet multiple entities by area.
`Middle`	"Undo" the most recent operation. Thus unblank the last pick, etc.
`Right`	Raise the full options menu for the selected object type, giving the option of performing any of the "quick" operations on it, regardless of the current mode.

In all cases the effect is immediate, for example clicking on a part to blank it results in the image being redrawn.

The **ENTITY** button can be used to access a popup via the right mouse button to change the default selection category from **PART** to any of the generic element classes that the model contains (PARTS, NODES, SOLIDS, SHELLS, THICK SHELLS BEAMS, SPRINGS etc). INCLUDES may also be selected if there is a `.ztf` file.

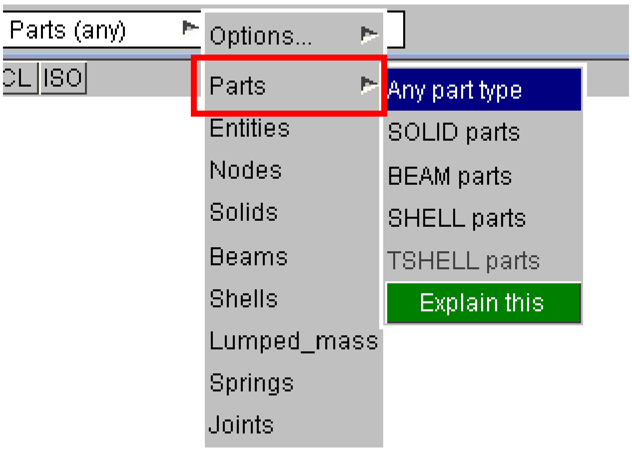


PART s are a special case:

The default setting for PART picking is to pick parts of any eligible element type.

But because the default screen picking process will tend to pick 2D and 3D elements (because they have a finite area), it can be difficult to pick **BEAM** parts if they overlay 2D or 3D mesh.

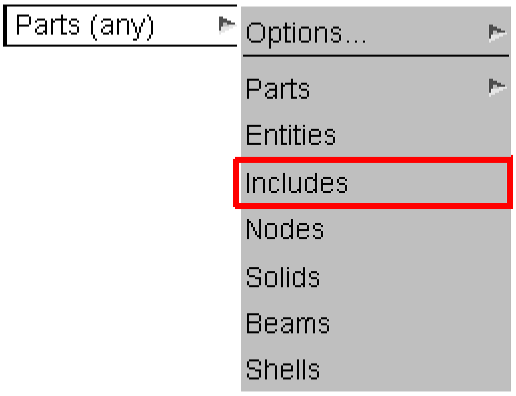
Therefore it is possible to restrict the type of part to be picked by underlying element type.



INCLUDE s are also a special case:

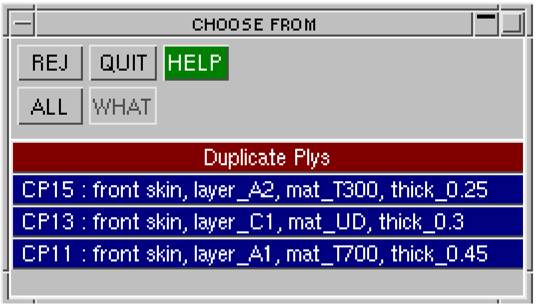
INCLUDE files are selected by PART, meaning the entities in the include file that contains the PART definition are selected. The selection of include files is recursive, so entities in any child include files are also selected.

If an entity is defined in one include file and the PART is defined in another both INCLUDE files are selected.



COMPOSITE PLY s are also a special case:

Because a shell can be in more than one composite ply, the screen pick is often ambiguous. If there is more than one ply where the user clicks, all relevant plys are listed in a popup box as shown in figure. The user may then:

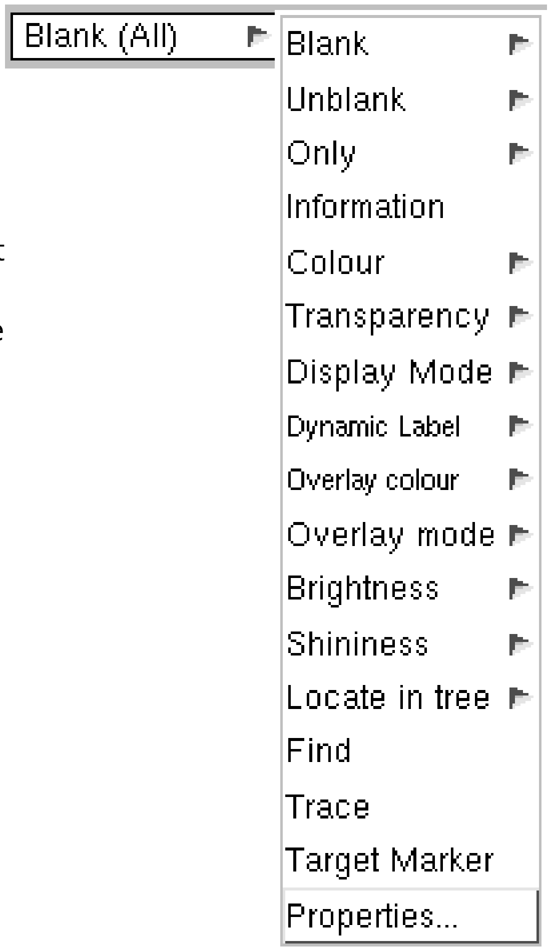


- Click on an item to select that ply,
- Select **ALL** the listed plys, or
- **REJ** ect the selection and return to screen pick.

If the user `drags` to select plys by area, all relevant plys are selected.

If the user `right` clicks to raise the options menu, the ply is automatically chosen (either as the topmost ply if layups are set-up, otherwise as the ply with the lowest ID).

In addition to being able to **BLANK** items the **OPERATION** menu can be used to select the operations listed above to apply to the items selected.



The **OPTION** button can then be used to select which windows a blank is applied in; and which colour, transparency level, display mode etc that is applied to the item when it is selected on the screen.

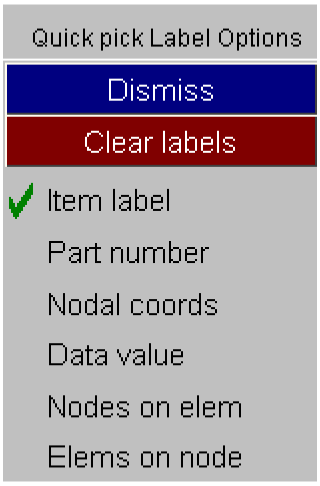
(Not all options will be available for parent item types.)

All operations carried out using these options and the mouse are stored and can be undone using the **MIDDLE** mouse button. Furthermore all of these options can be used while animating. To indicate which operation is currently active the mouse symbol will change as is appropriate.

6.5.1. Labelling

Labelling

Any permutation of of the following items can be selected for labelling, but note that some can only apply to nodes and some to elements.



Item Label	Draws the node or element label
Part Number	Draws the number of the part the item belongs to
Nodal coords	Draws the current [X,Y,Z] coordinates to the left of the node
Data value	Draws the data value associated with the currently visible plot data(eg CT,SI) to the left of the node position or element centre. A value of 0.0 is used if the currently visible plot does not imply data (e.g. LI, HI, SH drawing modes) or if there is an entity type/data plotting mismatch (e.g. beam element data froma contact surface data plot).
Nodes on elem	Draws the labels of all nodes on the selected element
Elems on node	Draws the labels of all elements attached to the selected node

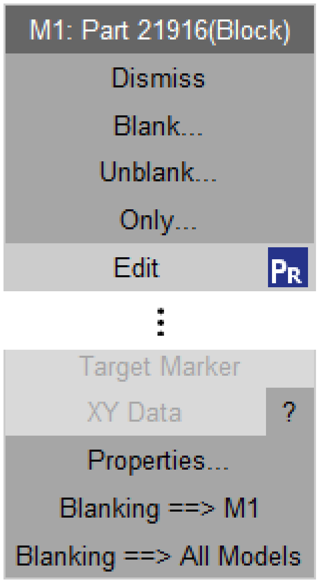
The labels persist during animation and redrawing, until the the **Clear Labels** option is selected.

6.5.2. Right-Click Quick-Pick

Right-Click Quick-Pick

When hovering the mouse pointer over a model pressing the right mouse button launches the quick-pick popup menu. The options available are the same as those described in ["QUICK PICK" Options] with the addition of two buttons to export blanking from the selected model to other windows and models. These options are:

Blanking ==> Mn	Exports blanking of selected model in window to same model in all active windows
Blanking ==> All Models	Exports blanking of selected model in window to all models in all active windows



6.6. "Tabs" for Multiple Graphics Windows

"Tabs" for multiple graphics windows

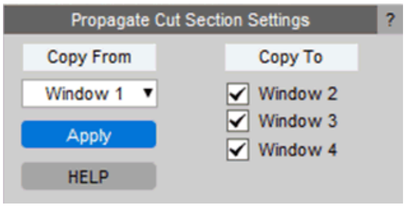
When more than one graphics window is in use most menu panels will have a "tab" button for each graphics window: **W1** , **W2** , etc. In this example there are four graphics windows.

These tabs control the graphics windows to which the commands issued in this panel apply: here any command would apply to all four windows.

The **==>** copies settings between windows immediately using the default window selections. By default the 1st active window is selected as the window to copy from and all the other active windows are selected as the windows to copy to.

The **...** displays a "Propagate" menu that can be used to copy settings between windows.

Multiple graphics windows are discussed in more detail in [Multiple Windows and Models] .

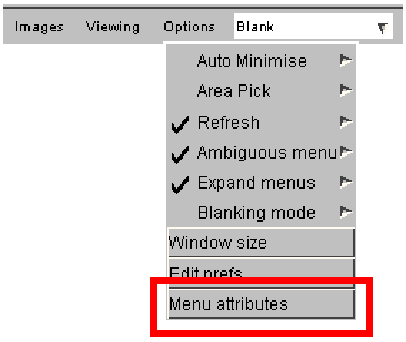


6.7. Customising the User Interface

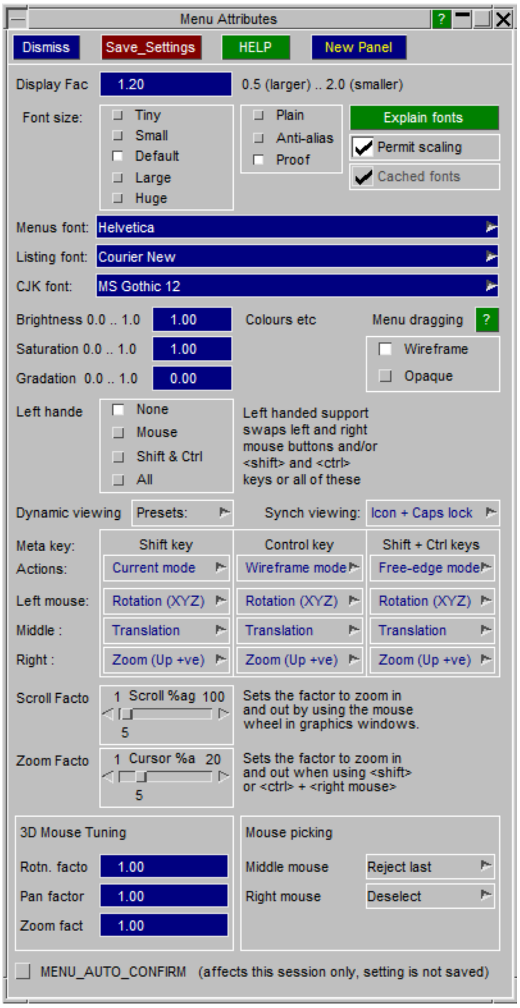
6.7.1. Customising Menu Size, Fonts, Colour and Mouse/Keyboard Behaviour

Customising Menu Size, Fonts, Colour and Mouse/Keyboard Behaviour

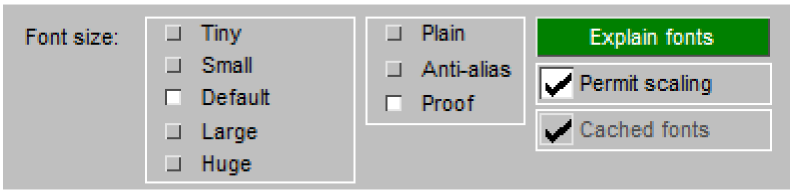
As mentioned in Mouse and Keyboard Usage for Screen Menu Interface, the scale of the menu interface, the font typeface and size, and also the left-handedness of the menu interface may be customised interactively using **Options > Menu Attributes.**

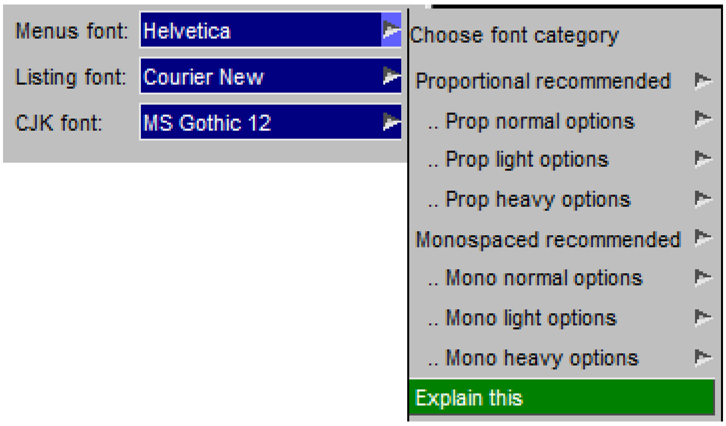


Gives the menu attributes panel:



Display Factor	Is a factor on the overall scale of the display, lying in the range 0.5 to 2.0, default 1.0. Larger values make the display seem bigger to the software, resulting in smaller menu panels and fonts. Smaller values increase the size of menu panels, buttons and fonts, and can be useful for the visually impaired. This factor can be especially useful on "wide screen" displays with very asymmetric horizontal and vertical resolutions. The operating system *should* determine the physical size of the display correctly. However we have observed a few instances where this does not happen, the symptoms being that fonts and menus appear either far too big or too small and cannot be corrected by using **Display Factor** . In this situation you may need to tell the software the physical dimensions of

	your display, and this process is described under [Setting the correct physical resolution for your display]() in the PRIMER manual.
Font size, quality and scaling	 On most displays the "Default" font size will give the best appearance in menu interface panels, but occasionally "Small" or "Large" fonts may look better. It is recommended that you set the Display Factor first in order to get the best overall layout on your display, then adjust the font size if necessary. Font quality has been improved in D3PLOT 17.0, and on most displays "Proof" quality will look best. However on low resolution displays it may look a little fuzzy due to the anti-aliasing process, and "Anti-alias" (coarser) or "Plain" (not anti-aliased) may give a crisper result. Font scaling ("Permit scaling") can be useful when your choice of font is a bit too large for the buttons in the user interface, since it allows the default font size to be reduced where text would overflow the space in a button. However it can result in a mixture of font sizes in a panel, which might improve legibility but looks untidy, so it is generally better to choose a Display Factor and Font size that work well together on your display, and turn scaling off. Cached fonts is an obscure setting that will only apply on Linux systems where the "core" X11 font package has not been loaded, and the software reverts to cached bitmaps. If you have font problems on Linux please contact Oasys Ltd for advice and help.

Font Typeface	 Historically D3PLOT only provided Helvetica, Times and Courier fonts, but in D3PLOT 17.0, a wider range of fonts has been made available. The default for the User interface is still Helvetica for menu panels (the "Menus font") and Courier for listings (the "Listing font"), but you can use the popup menus to select from any of the fonts on your computer. The range of fonts available will depend both on the operating system and what has been installed, but typically there can be many. To try to make the choice manageable these are separated into Proportionally spaced fonts, where character width varies. This is preferred for GUI panels with buttons. `Monospaced fonts, where each character width is the same.` This is preferred for text listings. Within each category fonts are also sorted by weight, with "normal" being the most commonly used. "Light" options tend to be narrower, permitting more characters to fit in a button, "Heavy" options tend to use bold text, and can be useful when using very large fonts - perhaps on a projector or when setting up the user interface for someone who is visually impaired.
Brightness Saturation	These affect the overall brightness and also the colour saturation of the user interface. They both lie in the range 0.0 to 1.0, default 1.0.
Left-Handed support	By default D3PLOT is set up for right-handed usage, which has influence on both mouse buttons and the keyboard "meta" keys: <shift> and <ctrl>. (The left and right meta keys have different functions during dynamic viewing: see dynamic viewing) You can swap the handedness of mouse and/or meta keys, which will reverse them in the left <=> right sense. **Note** : This swapping is local to D3PLOT, and is applied after any system user interface configuration. So if you configure your computer to swap mouse buttons globally, then swap them here, the net effect will be to have unswapped buttons again!

Dynamic viewing

By default D3PLOT uses the following dynamic viewing keyboard + mouse key actions:

Keyboard meta key	Viewing mode				Mouse button	Viewing action
<shift>	Normal	}		{	Left	Rotate in XY or Z
<ctrl>	Wireframe	}	+	{	Middle	Translate
<shift + ctrl>	Free edge	}		{	Right	Zoom (+ve upwards)

However different users have different tastes, and users who swap between different applications find it easier if they behave in similar ways. Therefore the following [permutations are available:

Viewing mode , may be assigned to keyboard meta-key(s) (ie <shift>, <ctrl> or <shift + ctrl>

Normal	will use the current display mode
Wireframe	only the line vectors in the current display mode
Free-edge	special "free edge lines only" display mode

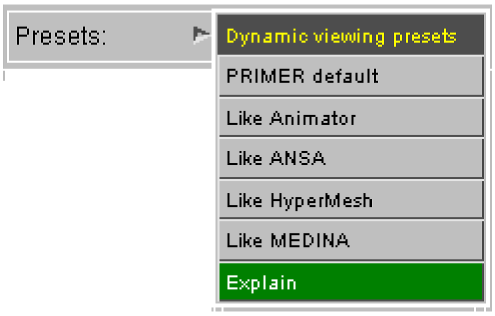
Dynamic rotation options , assigned to mouse buttons

Rotate XYZ	traditional D3PLOT behaviour, rotates in XY if cursor's initial position is in centre 2/3rd of screen, otherwise about Z
Rotate XY	rotates about screen XY only, regardless of where the cursor's initial position
Rotate Z	rotates about screen Z only, regardless of cursor initial position
Rotate Sphere	free rotation about any of XYZ, like grabbing a point in a virtual sphere and dragging it

Dynamic translation options, assigned to mouse buttons

Translate	model follows cursor movement in screen XY plane

Zoom options, assigned to mouse buttons

	Zoom (up +ve) up and to the right enlarge, down and to left reduce **Zoom (down +ve)** down and to the right enlarge, up and to left reduce
Presets	These preset options configure D3PLOT's dynamic viewing controls to operate in a similar way to those of the listed programmes. The descriptions "Like (program name)" are given only for ease of reference to certain combinations of key and mouse buttons used for dynamic viewing control.  ANIMATOR is a product of GNS mbH ANSA is a trademark of BETA CAE systems SA HYPERMESH is a registered trademark of Altair Engineering, Inc. MEDINA is a registered trademark of T-Systems GmbH The configurations these produce may not match exactly the actions in the given application, but they are the best that can be achieved at the present time with the options available.
Scroll factor	Determines the rate at which using the mouse scroll wheel to zoom in/out changes the image magnification factor. Smaller values will act more slowly, and larger ones more quickly - it is best set by experiment.
Zoom factor	Determines how rapidly the <meta key + mouse key> dynamic zoom operations above work. Again this is best set by trial and error.
3D Mouse tuning	Factors that are applied to translations/rotations when using a 3D mouse produced by 3DConnexion.
MENU_ AUTO_ CONFIRM	This is a special setting designed mainly for "batch" style usage, and it controls how "popup" windows that normally wait for acknowledgement from the user should respond. If it is switched on then these windows will assume that the user has clicked the default action (usually "OK") and continue operation without waiting. This can be useful when replaying scripts, but it is not recommended for normal interactive usage.

Saving Menu Attributes settings

The attributes above may be saved in the "oa_pref" file by using **Save_Settings** . Subsequent sessions of D3PLOT will pick these up and re-apply them. The "oa_pref" file is described in more detail in Appendix B .

For backwards compatibility these attributes may also be set using environment variables as described in Appendix M. Where conflicting settings exist those in the "oa_pref" file generated by the panel above (or by hand) will "win".

Note : The software potentially reads four "oa_pref" files when an application starts, in the following order:

(1) The OA_ADMIN_nn directory
(2) The OA_INSTALL directory
(3) The OA_HOME directory (default $HOME on Unix/Linux, %USERPROFILE% on Windows - typically C:\Documents and Settings\ *user_id*)
(4) The current directory (typically "Start in" directory on windows)

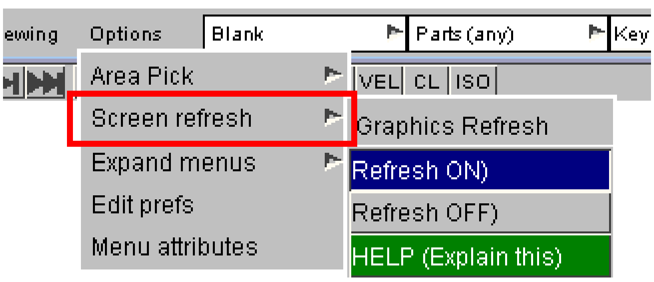
Save_Settings in this panel update the file (#2) above in OA_HOME, on the principle that you will have write permission there and - usually - it will not affect other users. However all "oa_pref" file settings are applied on the "last found wins" basis, so if you have file in your current directory with different settings these, being the last to be found, will "win".

6.7.2. Screen Refresh: Controlling Graphics Window Redraws

Screen Refresh: Controlling graphics window redraws

Normally graphics images in D3PLOT redraw at an acceptable speed, and the delay when refreshing "holes" left by menus popped up in front of the graphics window are not objectionable.

However if you are running a large model on a slow machine this may become a problem, and it is possible to turn off screen refreshes using **Options > Screen Refresh** . This will leave black holes when menu panels are unmapped, and you will have to issue an explicit redraw command to get rid of these. You can turn the refresh switch on again at any time.



You can save the graphics refresh status in the "oa_pref" file with the line:

```
d3plot*graphics_refresh : off | on
```

The default is "On".

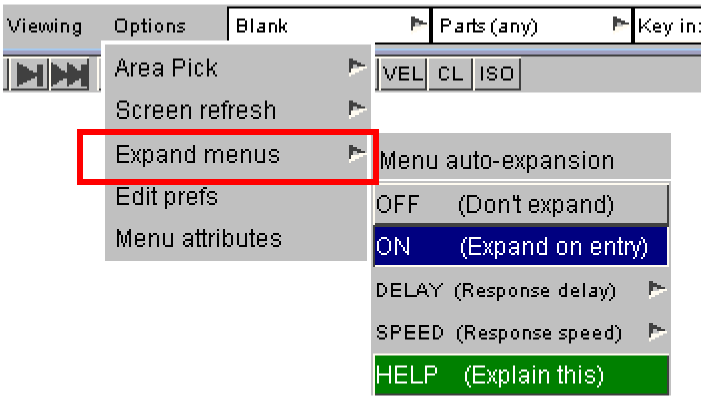
Note : Unlike PRIMER no "backing store" drawing is used in D3PLOT, so issues such as Bitmaps, Pixmaps and PBuffers do not arise.

6.7.3. Menu "Auto Expansion"

Menu "Auto Expansion"

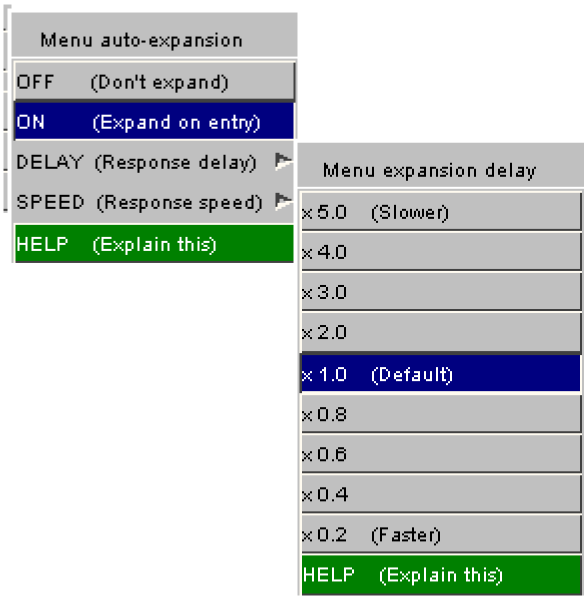
A few of the menus in D3PLOT are too narrow when first mapped to show all the columns of their data, so by default "auto expansion" is enabled. This causes the menus to widen themselves, typically to 90% of the enclosing width available, after a brief delay. You can control this behaviour using **Options > Expand Menus** as follows:

By default menu auto-expansion is **ON** , but you can suppress it by turning it **OFF**.



Controlling the speed and delay:

You can also control:



DELAY the time interval between the mouse entering a window, and the window starting to expand.

The delay time is controlled as a factor on the default behaviour.

The actual delay time will vary from system to system depending upon the Window system and underlying speed, but a typical delay will be approximately 0.5 seconds.

SPEED (Not shown here) is the rate at which the menu expands and contracts.

As above it is controlled as a factor on the default speed.

Saving Menu Auto Expansion Settings

The menu expansion parameters may be saved for future PRIMER sessions by setting the "oa_pref" file options:

```
d3plot*menu_expand:          ON | OFF
d3plot*menu_expand_delay:    Floating value in the range 0.1 ... 5.0
d3plot*menu_expand_speed:    Floating value in the range 0.1 ... 5.0
```

Full details of all "oa_pref" file options and environment variables are given in Appendix B

6.8. Shortcut Keys

Shortcut Keys

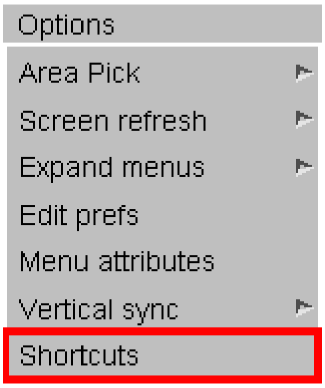
Some panels and actions can be accessed through pre-programmed shortcuts and from D3PLOT 9.4 onwards, the keys they are assigned to are customizable.

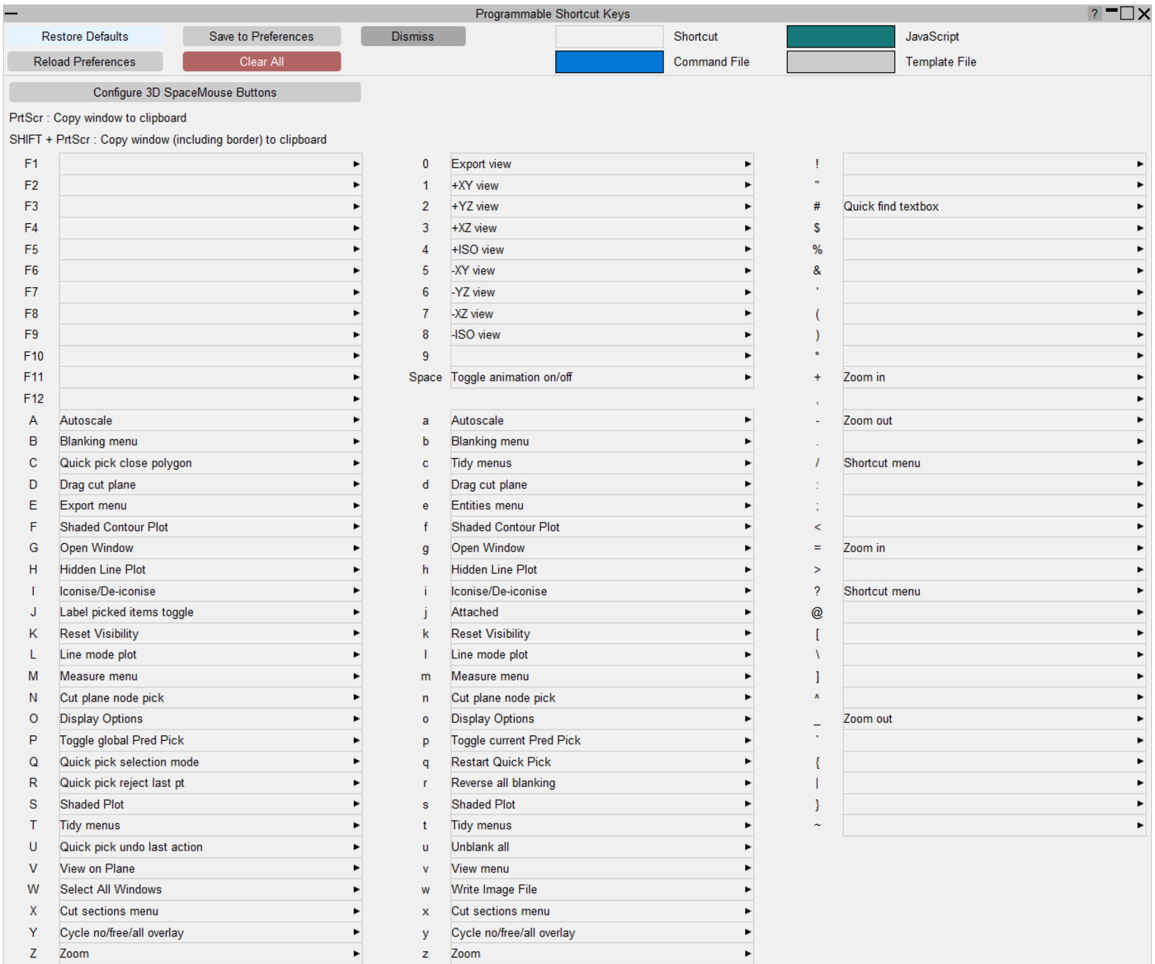
In D3PLOT 9.4, a number of new pre-programmed shortcuts have been added, including the top menu panels, all the contour buttons and the Lock and Centre buttons. JavaScripts and Command Files can also be assigned to a key.

A listing of the available shortcuts and the keys they are assigned to can be brought up by pressing either the '?' key (by default) or accessing it through the Options top menu.

This will bring up a panel, from which you may assign the shortcuts, JavaScripts and Command Files to the keys. Note that upper and lower case letters can be assigned different shortcuts.

A list of all the available pre-programmed shortcuts is given at the end of this section with their default key(s) if assigned.

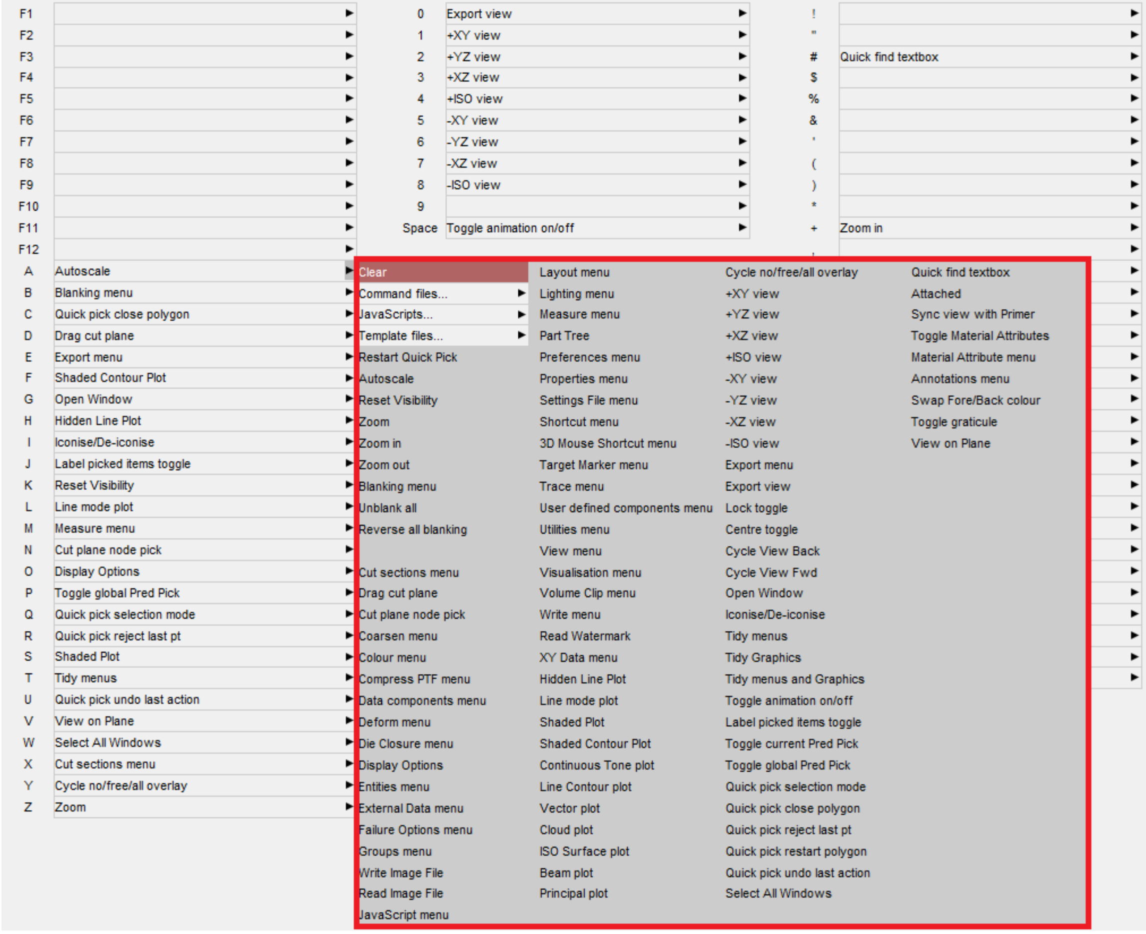




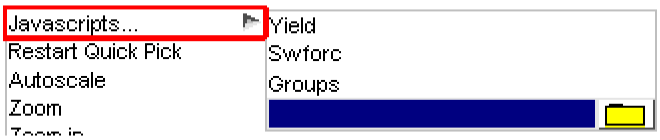
At the top of the panel you will see the following buttons.

Restore Defaults	Restores the shortcuts to their default keys, removing any shortcuts assigned by the user.
Save to Preferences	Saves the shortcuts to the oa_pref file in the home directory. They are saved in the format "d3plot*A_key: AUTOSCALE" where the first part defines which key the shortcut is assigned to and the second part is the shortcut being assigned. Each shortcut has a specific name to use in the oa_pref file, and a list is given below. When D3PLOT is started this is read and the saved shortcuts are restored.
Reload Preferences	Reloads the shortcuts from the oa_pref file in the home directory.
Clear All	Clears all the shortcuts on the panel.

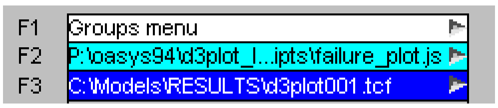
To assign a shortcut, right click on the key you want to assign it to. This will bring up a list of all available shortcuts in D3PLOT as well as the option to assign JavaScripts, Command Files and Template Files.



To assign a JavaScript or Command File to a key, right click on "JavaScripts..." or "Command files...". This will bring up another popup from which you can select the JavaScript or Command File. The popup will contain a list of Scripts that D3PLOT has picked up from the $OA_INSTALL and home directory. If the script you want is not in this list you can browse for it by clicking on the folder icon.



The listing of assigned keys is colour coded to easily distinguish between pre-programmed



shortcuts (white), JavaScripts (light-blue) and Command Files (dark-blue).

Pre-programmed Shortcuts: Defaults shown in bold, oa_pref name shown in brackets.

Plotting Modes

H/h – Hidden mode plot (HIDDEN)	**S/s** – Shaded mode plot (SHADED)
L/l – Line mode plot (LINE)	**F/f** – "Fringe" / SI plot (FRINGE)
Line contour plot (LINE_CONT)	Continuous Tone plot (CONT_TONE)
Vector plot (VECTOR_PLOT)	Cloud plot (CLOUD_PLOT)
ISO Surface plot (ISO_PLOT)	Beam plot (BEAM_PLOT)
Principal plot (PRINC_PLOT)	

View Controls

A/a – Autoscale current image (AUTOSCALE)	**3** – +XZ view (VIEW_P_XY)
v – View control panel (VIEW_MENU)	**4** – +ISO view (VIEW_P_ISO)
Y/y – Cycle through no/free/all overlay	**5** – -XY view (VIEW_N_XY)
Z/z – Zoom using cursor (ZOOM)	**6** – -YZ view (VIEW_N_YZ)
"+"/"=" – Zoom in (factor 2.0) (ZOOM_IN)	**7** – -XZ view (VIEW_N_XZ)
"-"/"_" – Zoom out (factor 0.5) (ZOOM_OUT)	**8** – -ISO view (VIEW_N_ISO)
1 – +XY view (VIEW_P_XY)	**0** – "Exports" the view of the current graphics window to all other active windows (EXPORT)
2 – +YZ view (VIEW_P_YZ)	Toggle Centre (CENTRE)
Toggle Lock (LOCK)	Cycle View Forward (CYCLE_VIEW_FWD)
Cycle View Back (CYCLE_VIEW_BACK)	

Blanking

`B/b` – Blanking control panel (BLANK)	j - Find attached (ATTACHED)
`R/r` – Reverse blanking of image (REVERSE)	
`U/u` – Unblank all (UNBLANK)	

Panels

`C/c` – Close all panels (TIDY_MENUS)	`O/o` – Overlay and Display panel (DISPLAY)
`D/d` – Drag cut plane (DRAG_CUT)	`P/p` – Properties panel (PROPERTIES)
`E` – Export menu (EXPORT_MENU)	`e` – Entity panel (ENTITIES)
`W/w` – Write image file panel (IMAGE_WRITE)	`M/m` – Measure panel (node -> node) (MEASURE)
`X/x` – Cut sections panel (CUT_SECTION)	`N/n` – Pick cut plane node(s) (CUT_PLANE)
`?/'/'` – Shortcut panel (SHORTCUT)	Coarsen panel (COARSEN)
Colour panel (COLOUR)	Compress panel (COMPRESS)
Die Closure panel (DIE_CLOSURE)	Data Components panel (DATA)
Deform panel (DEFORM)	External Data panel (EXTERNAL_DATA)
Failure Options panel (FAILURE)	Groups panel (GROUPS)
JavaScript panel (JAVA)	Layout panel (LAYOUT)
Lighting panel (LIGHTING)	Part Tree panel (PART_TREE)
Preferences panel (PREFERENCES)	Read Image file panel (IMAGE_READ)
Read Watermark panel (WATERMARK)	Settings File panel (SETTINGS)
Target Marker panel (TARGET)	Trace Node panel (TRACE)
User Defined Components panel (USER)	Utilities panel (UTILITIES)
Visualisation panel (VISUALISATION)	Volume Clip panel (VOL_CLIP)
Write panel (WRITE)	XY Data panel (XYDATA)

State Selection

`->` – Forward one state

`<-` – Backward one state

`<HOME>` - Jump to first state

`<END>` – Jump to last state

`<SHIFT> + ->` – Forward one frame

`<SHIFT> + <-` – Backward one frame

`<SPACE>` – Toggle animation (ANIMATE)

Windows

`G/g` – Open new window (NEW_WINDOW)

`T/t` – Tidy all windows (TIDY_MENUS)

`I/i` – Iconise windows (ICONISE)

`<DEL>` – In a graphics window erases dynamic labels

Miscellaneous

`Q/q` – Quit current pick action (QUICK_PICK)

V - View normal to plane for cut sections

`<PAGE DOWN>` – Move down a page

`<PAGE UP>` – Move up a page

J - Label picked items toggle (TOGGLE_PICK_LABEL)

Apply the current Attached Panel settings <ATTACHED>

If the mouse is in a graphics window, commands that imply a graphical change apply to that window only; otherwise they apply to all active graphic windows.

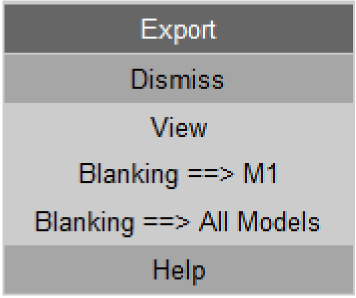
From D3PLOT 11.0 onwards, there is an option to configure 3D mouse buttons through the shortcuts panel. Click on the **Configure 3D SpaceMouse buttons** to assign functions, macros and JavaScripts to buttons on a 3DConnexion 3D mouse. See 3D Mouse for more information on 3D mice.

6.8.1. Export Menu

Export Menu

The shortcut key "E" or right-clicking the mouse in an empty part of a window launches the Export Menu in that window. The Export Menu offers the following options:

View	Exports view of current window to other active windows
Blanking ==> Mn	Exports blanking of model in window to same model in all active windows
Blanking ==> All Models	Exports blanking of model in window to all models in all active windows



6.9. Predictive Picking and Menu "Hover Over"

Predictive Picking and Menu "Hover Over"

"Predictive picking" highlights what would be picked were you to left-click with the mouse.

"Menu Hover Over" highlights items in menu lists, helping you to identify what they are.

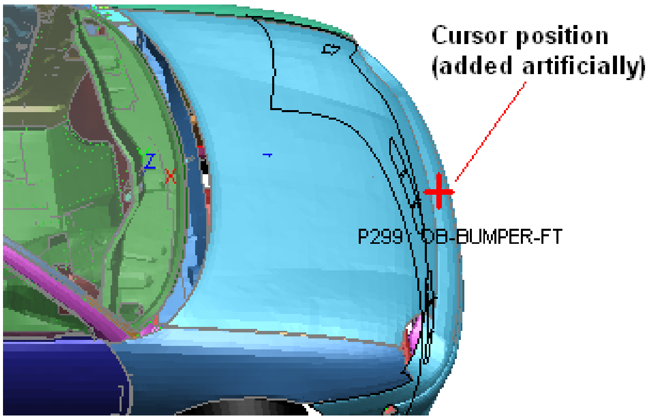
6.9.1. Description of Predictive Picking

Description of Predictive picking

From D3PLOT 10.0 onwards all screen-picking operations have "predictive picking" enabled by default. This means that when you move the cursor into the graphics window and position it over something pickable in the current context, the item in question will be highlighted by sketching and labelling it, identifying what would be selected were you to perform a left mouse click at that position.

In this example the cursor (red cross added artificially here) has been hovered over the front bumper of a vehicle model.

The current mode is the default "Quick pick by part", so the part making up the bumper has been sketched in free edge mode, and labelled with its id and title, here "P299 OB-BUMPER-FT".



(Part and other item titles will only be available if you have read a ZTF file generated by PRIMER. In their absence only the label will be shown.)

The sketching used to highlight items is transient: it will disappear as you move the cursor away from the object in question, and there is no need to refresh the graphics window to get rid of it.

In the example here the current pick mode was "Quick pick by part". Predictive picking is always associated with the current picking operation, so for example if you chose **[Blank] Shell** then the current picking mode would be to select a shell, and predictive picking would change to highlighting shells under the cursor.

6.9.2. Controlling Predictive Picking

Controlling Predictive Picking

Most of the time Predictive Picking is helpful, but there are two situations in which you might want to turn it off:

1. If your computer is very slow, or you are displaying graphics over a network, you may find that the need to keep updating the display as the cursor position moves makes the response sluggish.
2. If your image is very complex, and you are picking items which generate a lot of extra graphics when they are highlighted (typically sets, or contacts defined by set) you may find that predictive pick highlighting becomes a nuisance.

In the first situation you might want to turn it off for all picking operations; but in the second you may just want to suppress it for the duration of the current pick operation, turning it back on when you revert to picking items that are less visually complex. Therefore two levels of control are provided:

Switching on/off temporarily for this picking operation only

The **[PP]** button to the right of the "Quick Pick" selection buttons can be used to toggle predictive picking on/off ***for the current picking operation only.***

As an alternative you can use the " **p** " (note lower case) keyboard short-cut to have exactly the same effect.

This only affects the current picking operation, and the setting is "forgotten" once that operation ends.

Switching on/off globally

Programme-wide predictive picking can be toggled on/off using the " **P** " (note upper case) keyboard short cut.

This is not "remembered" automatically, so a future D3PLOT session will default to the standard setting of predictive picking being globally active

Saving changes to predictive picking settings

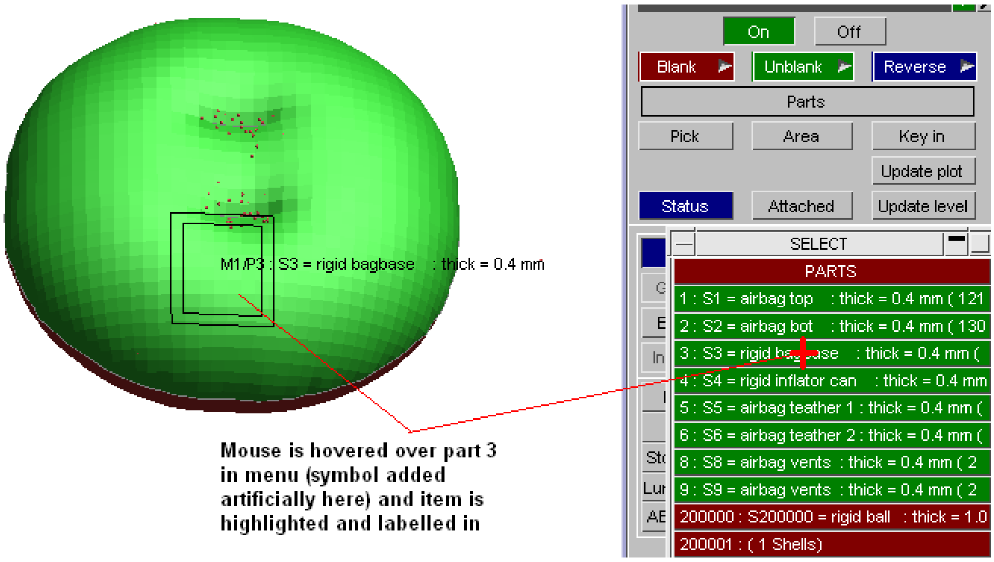
The effect of the "P" shortcut is not "remembered" automatically, so a future D3PLOT session will default to the standard setting of predictive picking being globally active. If you want to set something other than the default status this can be done via the following two oa_pref file options:

d3plot*predictive_pick:	**ON** or **OFF**	Whether predictive picking is active at all
d3plot*predictive_label:	**ON** or **OFF**	When predictive picking is active whether or not it also labels the items being sketched.

6.9.3. Description of Menu "Hover Over" Highlighting

Description of Menu "hover over" highlighting

Menu "Hover over" highlighting is very similar to Predictive picking. Whenever D3PLOT builds a menu showing a list of items for selection then hovering the cursor over a menu row will highlight and label that item on the screen.



Changing menu hover-over settings

By default "menu hover-over" is active, but you can change these settings with the oa_pref options:

`d3plot*menu_sketch:`	**ON** or **OFF**	Whether menu hover-over is active at all
`d3plot*menu_label:`	**ON** or **OFF**	When hover-over is active whether or not it also labels the items being sketched.

6.10. Colours

Colours

The colour popup allows users to select a standard colour or set-up and use a user-defined colour.

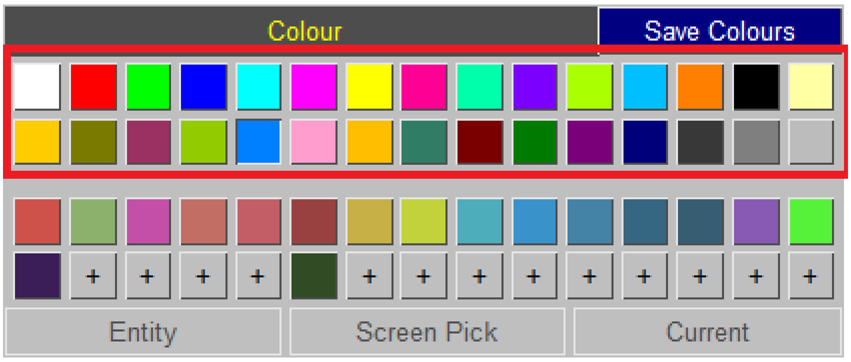
For some menus special context colours are available, for example "Entity", "Default" or "Background". These options are explained in more detail in the sections of the manual about that menu.

The Properties panel, Part Tree and Quick Pick menus allow colours by model and by include. For parts and elements they also allow colours by material, by section or by part tree assembly. If that information is not available to D3PLOT, in particular if there is no ZTF file, then the items will be shown in grey.

6.10.1. Standard Core Colours

Standard Core Colours

The top two rows show the 30 standard core colours.

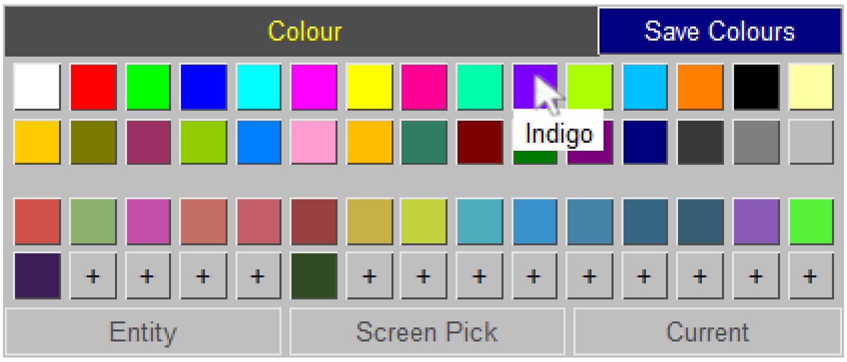


When you hover over the colour the name is shown.

This name can be used to specify this colour in preferences and dialogue inputs.

In T/HIS this name can be used in JavaScript and FAST-TCF.

When using the name, "_" is used instead of " ", for example "Hot Pink" becomes "HOT_PINK".



The standard core colours available very similar in D3PLOT and T/HIS. The following colours are a similar shade but have different names:

D3PLOT	**T/HIS**
Red/Magenta	Orange
Green/Cyan	Turquoise
Yellow/Green	Lime
Light Blue	Sky
Dark Orange	Pink
Cyan/Blue	Medium Blue
Red/Orange	Light Pink
Grey	Medium Grey

6.10.2. User Defined Colours

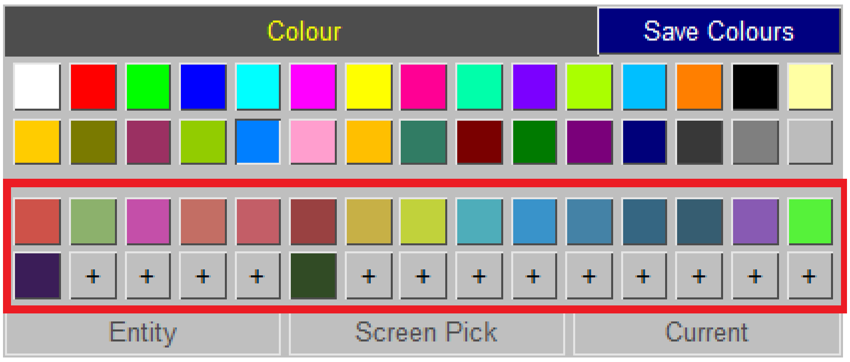
User-Defined Colours

The lower rows show the user-defined colours. There can be up to 150 user-defined colours.

Click on a user-defined colour to apply it, or click on an empty slot to create a new user-defined colour.

User-defined colours can be used in the dialogue input by specifying their name.

In T/HIS user-defined colours can be used in FAST-TCF.



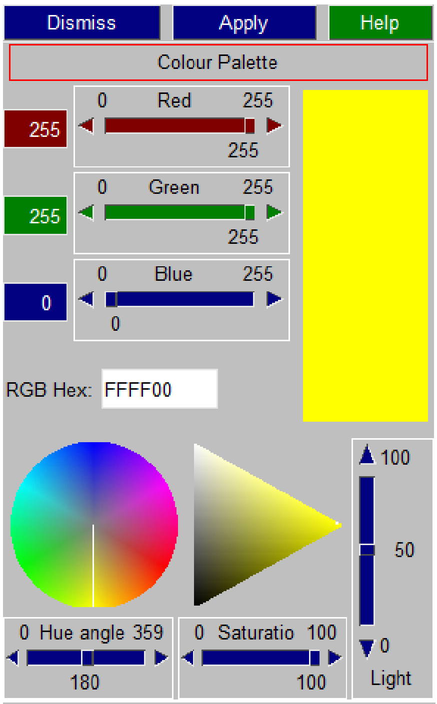
Creating

To create a new user colour click on an empty slot. This maps a colour palette.

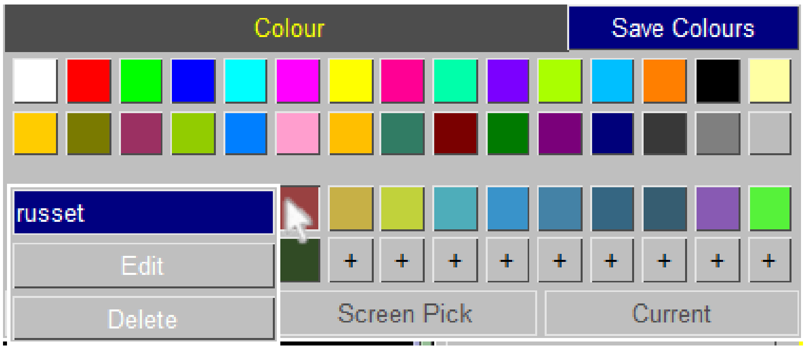
The colour can be edited a number of ways:

- Using sliders to set the red, green and blue value,
- Inputting a hex colour code,
- Clicking on the colour wheel and cone, or
- Using sliders to set the hue, light and saturation levels.

When you create a colour it is applied.



Editing



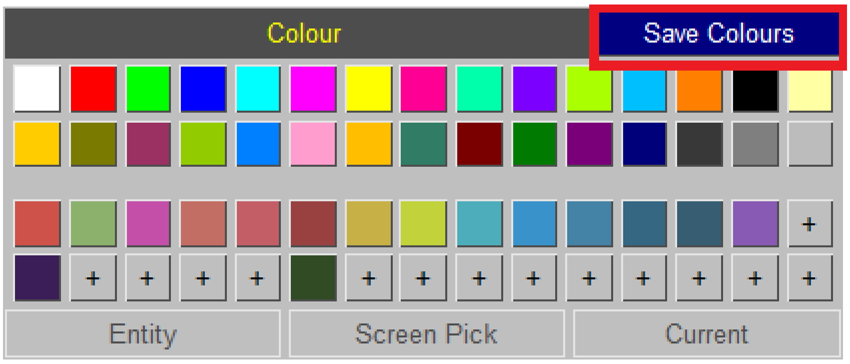
Hover over a user colour to edit it. You have the choice to change the name of the colour, **Edit** it, or **Delete** it.

The user-defined colours are given the standard name, for example "user_1". They can be renamed. The name must start with a letter and gets set to all lower case. If the name is not unique, a number will be appended to it, for example "green_1".

Edit maps the colour palette. If you edit a colour it is then applied.

Delete removes a colour. The colour is no longer available when you next open the colour popup.

Saving



The user-defined colours can be saved. The same user-defined colour are then available when you next run D3PLOT or T/HIS.

The user-defined colours are stored in the user_colours.xml file. If the user has permission to modify things in the INSTALL directory, the user is given the option to either save the user colours to the INSTALL directory (which is sometimes visible to multiple users) or their HOME directory.

Alternatively, the preference `user_colour_file` can be set to specify an .xml file.

When D3PLOT or T/HIS is next started the user_colours.xml file is read in.

If the same colour, for example "user_1", is defined in the user_colours.xml file in both the INSTALL and HOME directory, the HOME directory user_colours.xml file takes precedence.

If the preference `user_colour_file` has been set, any user_colours.xml file in the HOME directory is ignored. If a colour is also defined in the user_colours.xml file in the INSTALL directory, the `user_colour_file` .xml file takes precedence.

For T/HIS, if a user colour was previously set-up using a preference, for example **`this*user_colour1`** , and that colour slot is also defined in a user_colours.xml file, the user_colours.xml file takes precedence.

6.10.3. T/HIS Link

T/HIS Link

When running the T/HIS link any user colours created in D3PLOT (or in T/HIS) will be available in the other program. When T/HIS is first opened it sets-up the user colours to match the current D3PLOT session, rather than using a saved user_colours.xml file.

7. Basic Data Extraction and Plotting

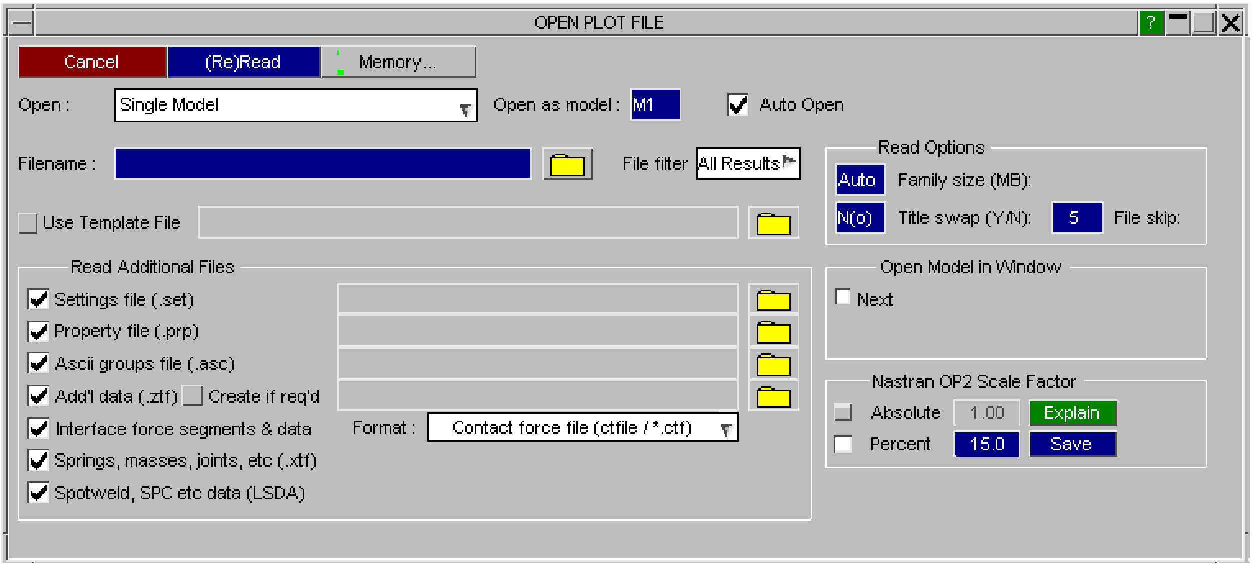
BASIC DATA EXTRACTION AND PLOTTING

This section describes how to extract data from disk, and how to display it graphically. Screen menu usage is assumed, although brief references will be made to equivalent command-line instructions where appropriate.

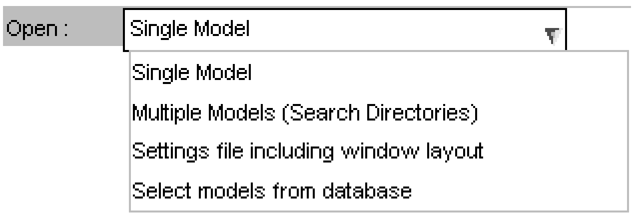
7.1. Reading Results

Reading Results

The Open Model panel is shown below.



By default this panel will allow you to select a single model and open it (see [Open a Single Model](#))



Alternatively this panel can be used to either:

(i) Search directories for results and open open multiple models (see Search Directories Recursively)

(ii) Read a settings file containing model information (see Settings File Including Window Layout).

(iii) Open a model database and select the models you want to read (see Select Models From Database)

Cancel

Dismisses the panel without reading a model in to D3PLOT.

(Re)Read

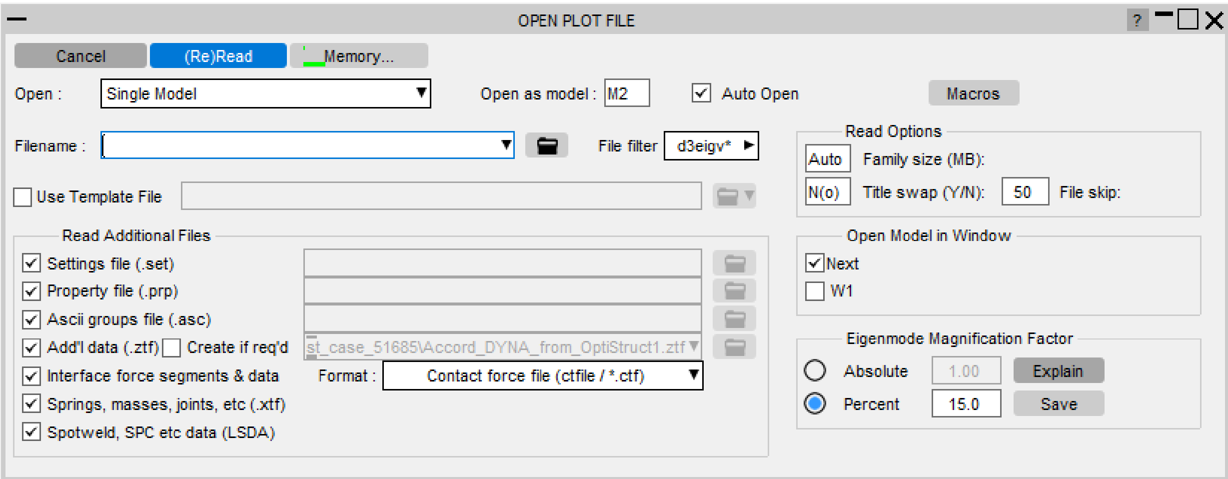
Reads a model in to D3PLOT.

MEMORY...

Maps the standard memory management box (see MEMORY Viewing and Controlling the Memory Usage for this Process and the Whole Machine). This allows you to set database memory limits and display mode before opening the file, which may be necessary in exceptional circumstances.

7.1.1. Open a Single Model

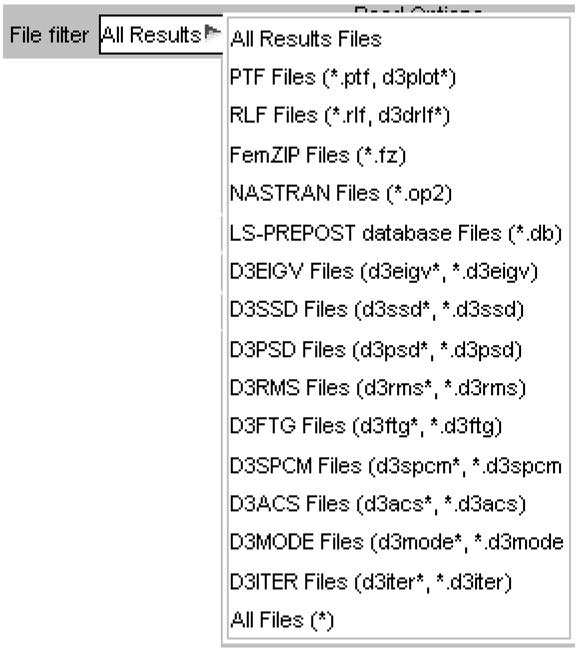
Open a Single Model



Filename

You can type a filename into the text entry box or use the  button to obtain a standard file selector box .

File Filter



The **File filter** button controls what extension will be used to search for file types in the file selector box.

D3PLOT requires a complete state, or equivalent, file in order to run. Under Oasys Ltd conventions this will have the filename format **<name>.ptf** , but any name is acceptable so long as the contents are recognisable.

The default filter box "pattern" is set to **All Results Files** so that all available results files will be shown in the file selector. (See LS-DYNA output files processed and Other Output Files Processed for a list of supported file types and names.)

To make using the standard file filter box easier you can pre-select the "pattern" that will be used for scanning files on disk using the options shown here. Of course any pattern can still be typed into the file filter box itself.

Use Template File

From D3PLOT 10.0 onwards, D3PLOT can read an optional Template file that contains information on which windows models are located in and setting for model offsets, colours and initial plotting modes. If a template file is selected then the options to control which windows models are located in will be disabled.

For more information on the format of template file and the options it contains see Template File .

Read Options

Family size (MB) :

The binary output files written by LS-DYNA form "families". Each family has a root member and may have children. The maximum size of any member of a family is set when LS-DYNA runs, the default being 7MB, however any size > 1MB can be used.

By default D3PLOT determines the family member size automatically (the **Auto** setting shown here), but you can override this by entering a size in MB. This is almost never necessary: read Disk Format of Binary Database Files before doing this.

File skip:

File families should form a contiguous sequence (root, member #1, member #2, ...) But it is sometimes the case that members are missing: intermediate members be deleted to save disk space, and occasionally LS-DYNA skips members.

The **File skip** value (here zero) is the number of missing members that D3PLOT will skip before giving up its search and deciding that it has found the end of a file family. See Disk Format of Binary Database Files for further information.

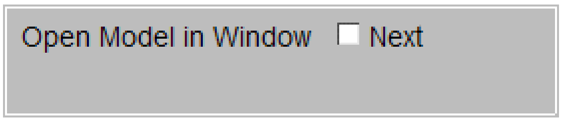
Title swap:

It is unfortunately the case that some versions of LS-DYNA have been compiled with numeric conversion flags which endian swap their output. This works fine for numbers, but scrambles the title (the string ABCDEFGH becomes DCBAHGFE).

If your title looks like garbage try changing this field to **Y(es)** to see if this fixes the problem. (Note that you can do this at any time during a D3PLOT session: see The FILE > popup menu options).

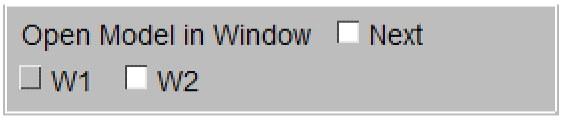
Open Model in Window

By Default D3PLOT will open each model in a new Graphics Window.

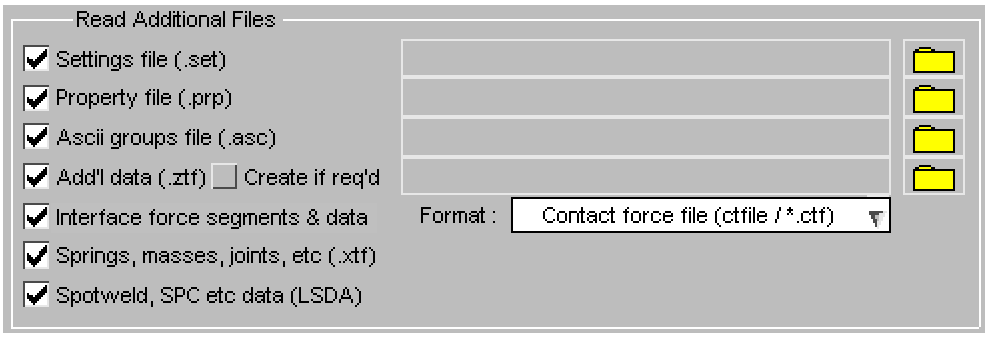


If a second or subsequent model is opened D3PLOT will offer the choice of opening the model in a new Window (**Next**) or one of the existing Windows (**W1** , **W2** ...).

If more than one Window is selected then D3PLOT will add the model to each of the Windows.



Read Additional Files



In addition to the main model file D3PLOT can read a number of additional files. By default all of these additional files will be selected to be read.

If the Auto Open option is selected then D3PLOT will automatically search for these additional files in the model directory and load any that are found.

Model Independent Files

Settings file (.set)	`d3plotnnn.set` or `<jobname>.set`	Contains information about programme settings on a per-window basis, such as background colour, cut-sections, data component, etc. These will be read if found and will automatically create new windows if required, and set them up as they were before.
Property file (.prp)	`<` ***`filename`*** `>_nnn.prp` or `<jobname>.prp`	Contain model-related information such as colour, transparency, overlay, etc that has previously been written from the **PROPS** panel. These files will be reread if found so that all this status information is automatically restored.
Ascii groups file (.asc)	`group001.asc`	Contains optional group information in a human-readable form which matches that used by PRIMER in the keyword input deck. If such a file is read it will supersede those groups currently stored for this model

Settings and Properties files are described more fully under **UTILITIES, SETTINGS_FILE**.

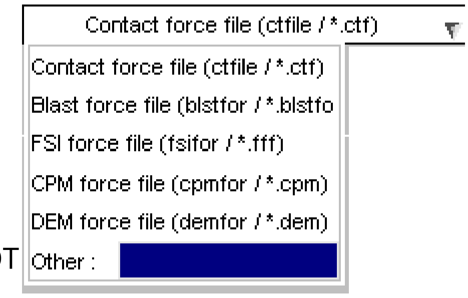
Group handling in D3PLOT 9.0 was extensively modified and improved, and it is described more fully under **GROUPS**.

Model Specific Files

Add'l data (ztf) The ZTF file is created by PRIMER and it contains a lot of additional data that D3PLOT can use to plot entities and data that is not included in the main model files. All of the data that was previously written to the XTF file by the SMP version of LS-DYNA that D3PLOT uses to draw springs, joints, stonewalls etc is also written to the ZTF file.

Interface force segments and data

LS-DYNA can output interface force files with different names depending on the analysis type (see LS-DYNA output files processed). The popup can be used to select which one D3PLOT should try to read.



In a model with many contact surfaces omitting the interface force file can speed up graphics and save memory, since contact segments will be ignored.

Springs, masses, joints (xtf)

Contains information on springs, joints stonewalls and lumped masses. Omitting the `XTF` file contents will not give similar savings to those obtained by omitting the CTF file and is not generally recommended. MPP versions of LS-DYNA do not write a `XTF` file, so from D3PLOT 9.0 onwards, the equivalent data is also present in the ZTF file.

Spotweld, SPC etc (LSDA)

From D3PLOT 9.4 onwards, D3PLOT can also read some additional data from the LSDA (binout) file. If your model contains spotwelds, springs, seatbelts or restrained nodes then D3PLOT will be able to plot come data components for these items if you have turned on output for them to the LSDA file.

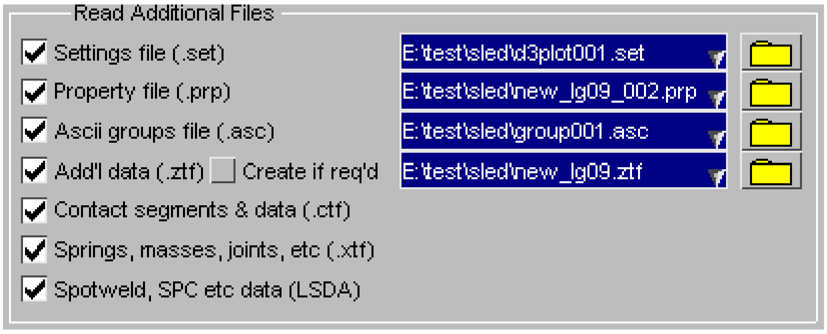
It is recommended that both `XTF` and `ZTF` files should always be read if present.

Details of the contents of all these files are given in Contents of the LS-DYNA Database Files Processed by D3PLOT

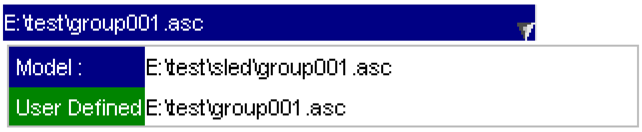
Manually Selecting Additional Files

If the 

option is deselected then once a model has been selected D3PLOT will search for any additional files and display any that it finds.

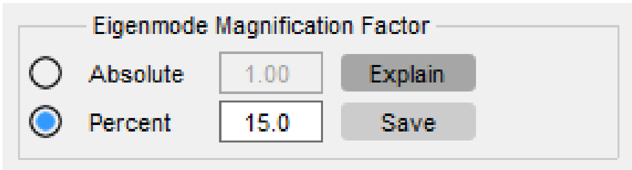


For the **.set** , **.prp** , **.asc** , and **.ztf** the text boxes and file selectors can be used to select alternative files if required.



After selecting an alternative file you can switch between the automatically found file and the user defined one using the popup menu attached to the text box.

Eigenmode Magnification Factor



When reading in modal analysis results (e.g. Nastran, OptiStruct or Frequency Domain Files in D3PLOT such as d3eigv), a magnification will automatically be applied to scale the displacements. This is done because deformations are generally relatively small for these analyses. Further magnifications make the mode shape more visible and easier to analyse. There are two magnification setting types: "absolute" and "percentage". Setting the type to "absolute" sets the same magnification factor to each mode shape and setting it to "percent" will set the maximum displacement for each mode shape to be a percentage of the model size. This value can later be modified in the Deform -> Magnify Panel.

The "Save" button will save the settings to your oa_pref file for use in future sessions.

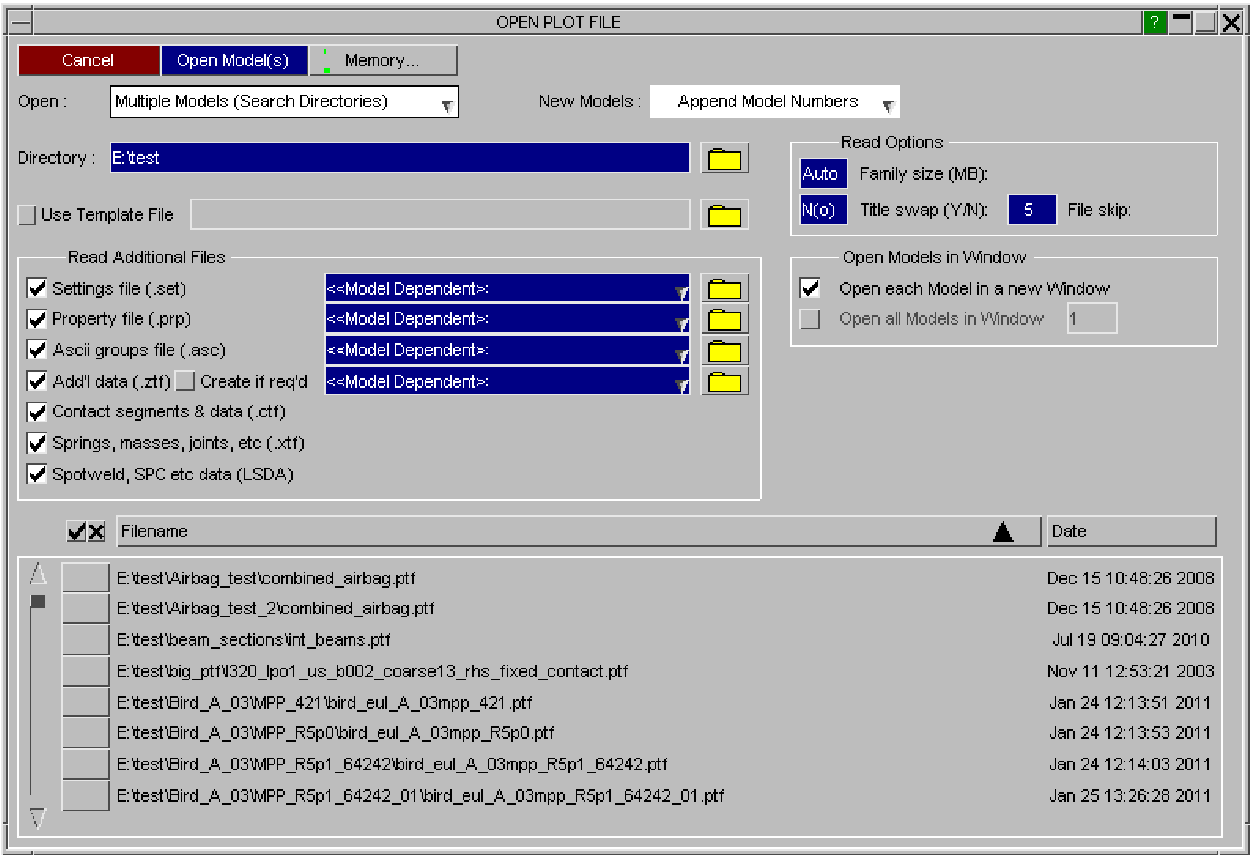
7.1.2. Search Directories Recursively

Search Directories Recursively

Multiple models can be opened by using the option to search directories recursively.

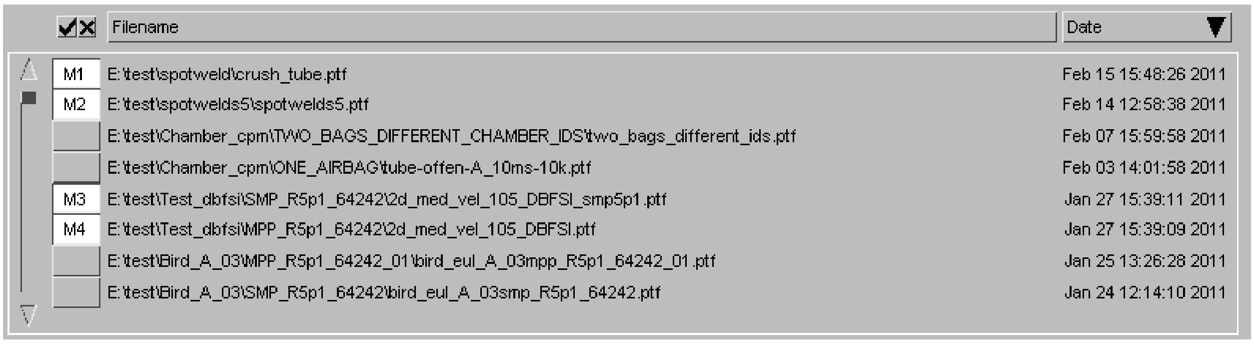
After a directory has been specified D3PLOT will display a list of all the models it can find in the directory structure and each file can be selected

If the user selects more than 32 models or if the number selected + any models already loaded into D3PLOT is greater than 32 then D3PLOT will open the models selected until the limit of 32 is reached.

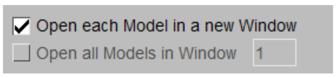


The list of models can be sorted by either alphabetically by directory name or by date into either ascending or descending order.

As each model is selected the model number that it will be read in as will be displayed.



Open Models in Window



When multiple models are read each model can either be read into a separate Window or all of the models can be loaded into an existing Window

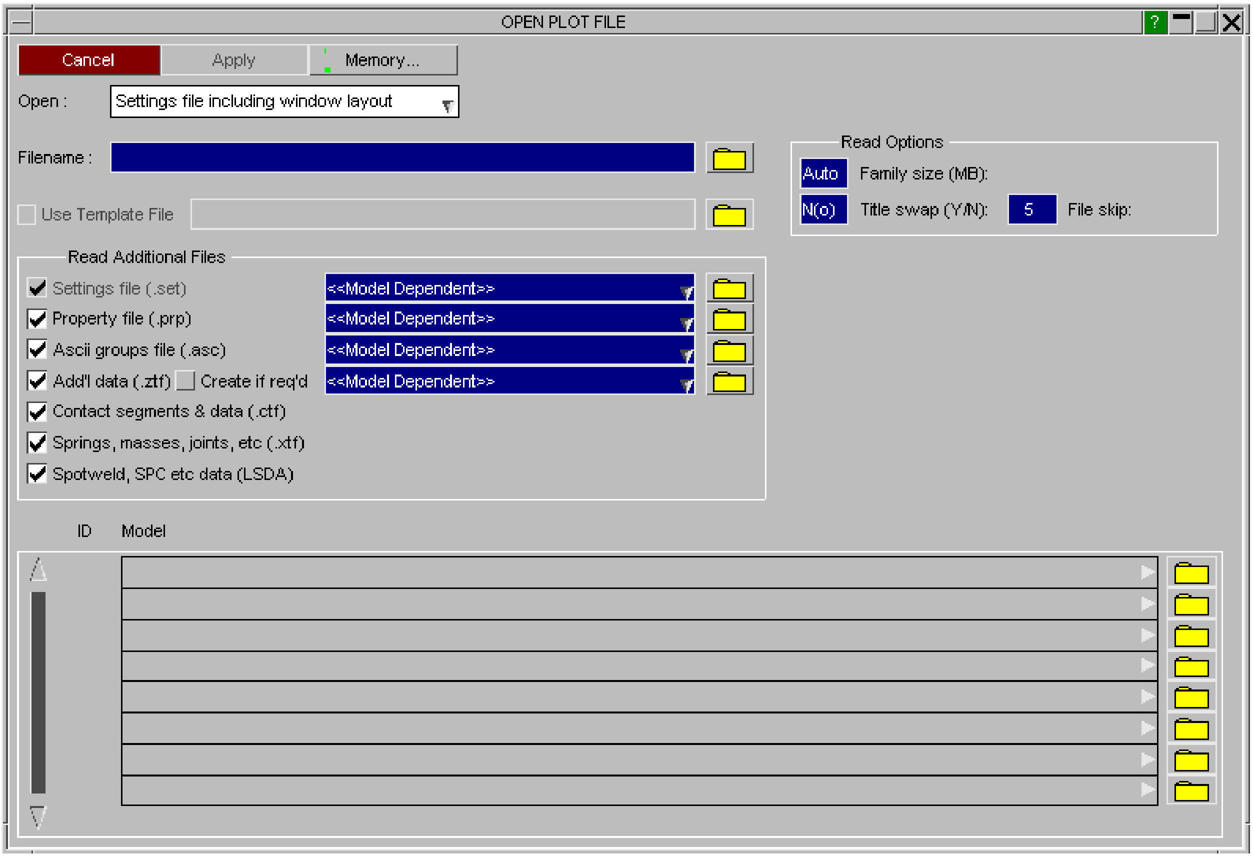
7.1.3. Settings File Including Window Layout

Settings File Including Window Layout

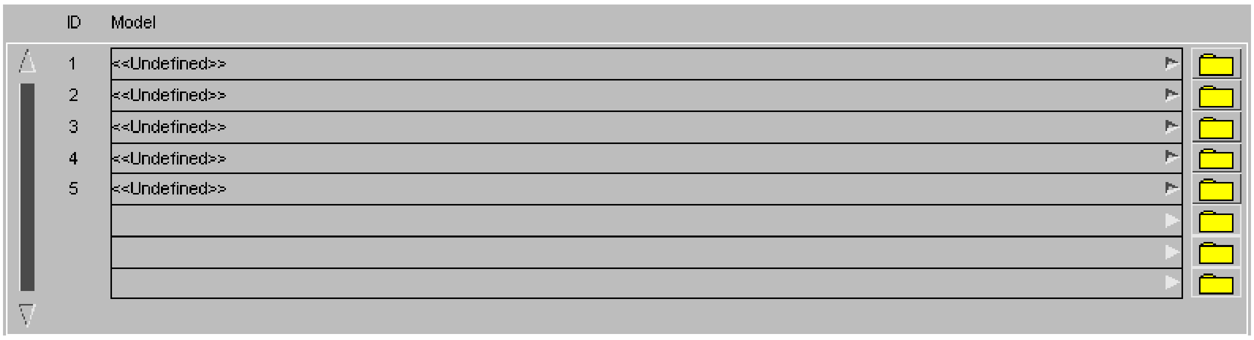
From D3PLOT 9.3 onwards a D3PLOT settings file contains information on the number of models that were open when the file was saved along with the window and page layout. The filenames for the models are not stored in the settings file, just the number of models and which model (M1, M2 ...) was located in each window.

If the setting file was saved while the D3PLOT->T/HIS link (see T/HIS the D3PLOT <=> T/HIS link) was running then the setting file will also contain information of the number of T/HIS graphs and it will contain a T/HIS FAST-TCF script which will regenerate the graphs contents.

This option can be used to reload a D3PLOT 9.3 settings file and to restore the model and Window layout.

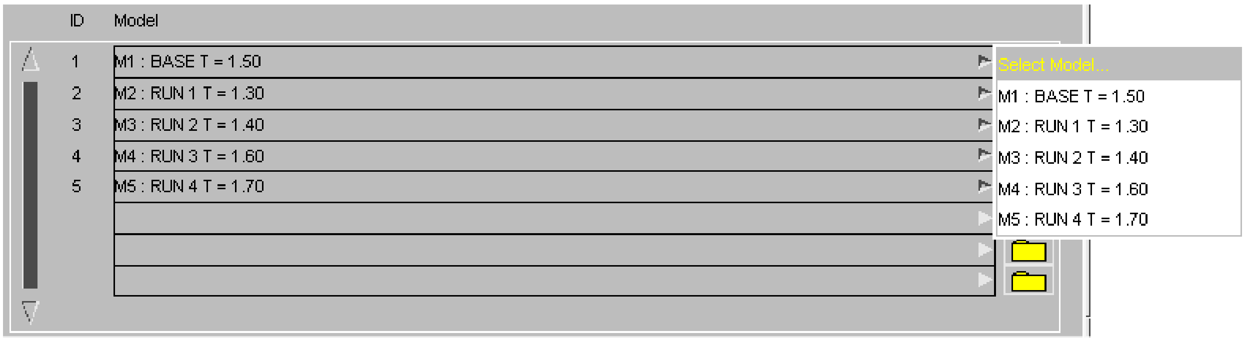


After a settings file is selected it's contents will be scanned to see how many models are required and D3PLOT will then display a list to allow the models to be selected.



If models have already been read into D3PLOT they will be automatically selected for the models to use when replaying the settings file.

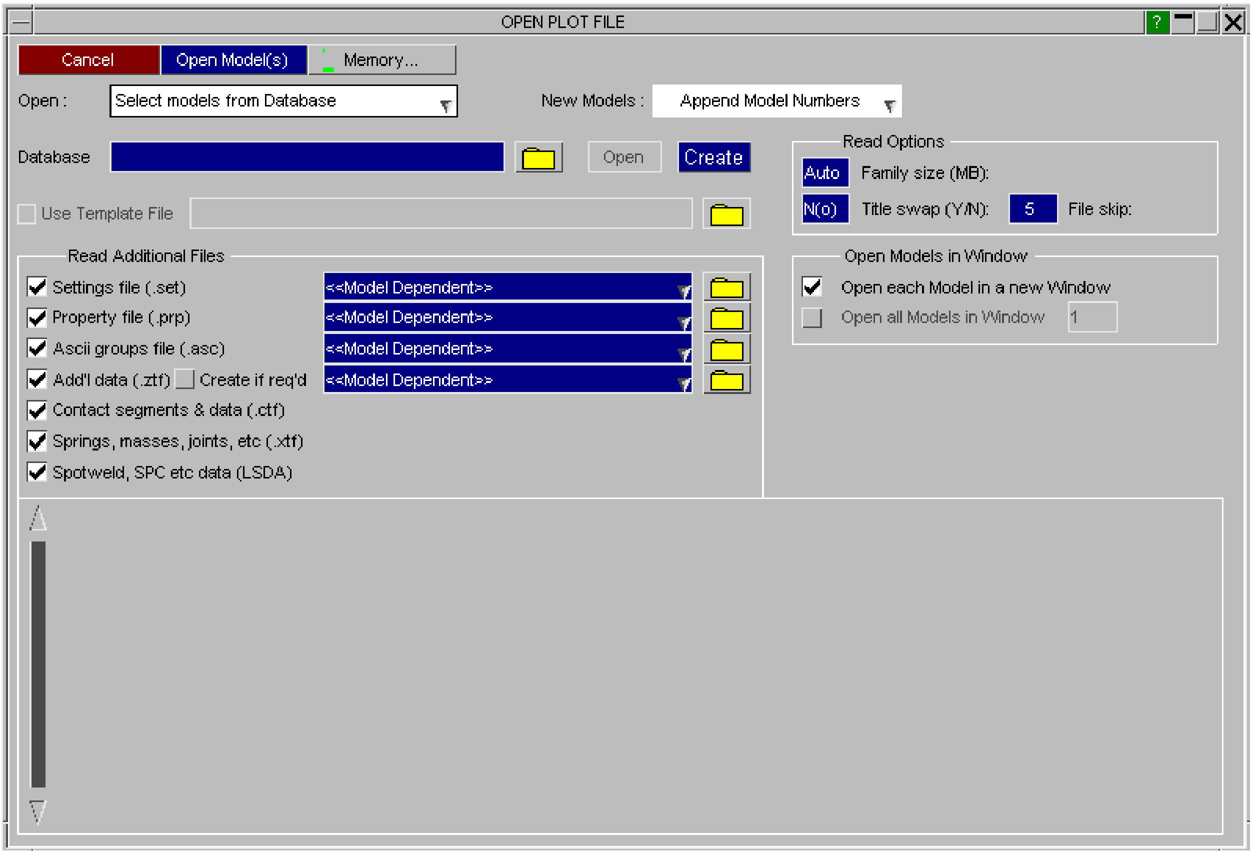
Any of the pre-selected models can be changed by using the popup menu to select a different model.



7.1.4. Select Models From Database

Select Models From Database

From D3PLOT 10.0 onwards, D3PLOT can select models from a model database. The database file is an XML format file that contains information on where models are located along with a brief description of each model, (see Database Format for more details on the file format)



To select a model database either enter it's name in the text box or use the file selector.

The default model database can be specified as a command line argument (see Command Line Options for more details). The default database filename and location can also be specified in the preference file (see Appendix B for more details)

```
d3plot*database_dir:
d3plot*database_file:
```

After a database file has been selected it's contents will be read and D3PLOT will display a Tree Like menu showing the contents of the database.

As each item is displayed D3PLOT will check to see if the files that it refers to exist.

If a file does exist then a green tick will be displayed

If a file does not exist then a red cross will be displayed

The number of levels in the database that are automatically expanded when it is first displayed can be specified in the preference file (see Appendix B for more details)

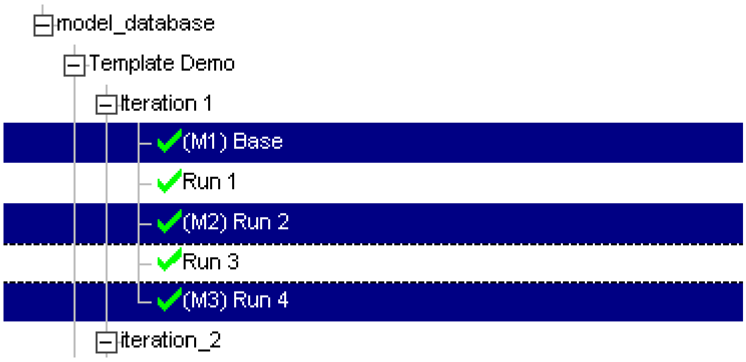
```
d3plot*database_expand:
```

After selecting the required models use **Apply** to close the database window and return to the main menu where the selected models will be displayed along with the model numbers they will be read in as.

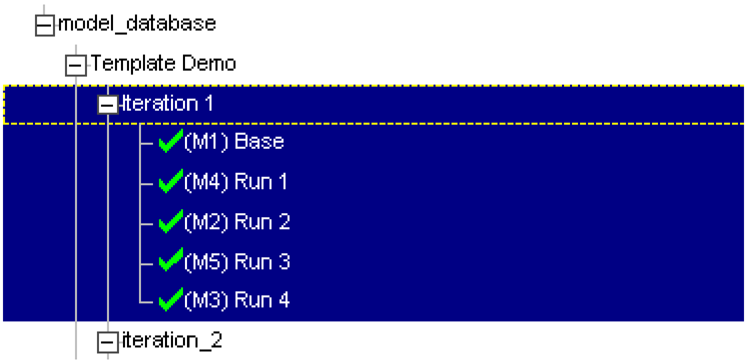
Selecting Models

Models can be selected and deselected by clicking on each row. Multiple model can be selected by clicking on the 1st model and holding down SHIFT while selecting the last model in the range.

As each model is selected the model number than it will be read in as is automatically displayed alongside the model description.



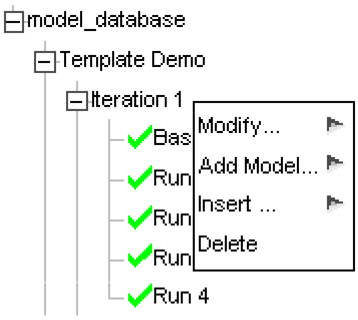
A complete branch can be selected/deselected by selecting the branch label (Iteration 1).



Modifying the Database

Database entries can be added, removed and modified by right clicking on a branch label or a model description

Right clicking on a branch label will display 4 options



Modify ... Modify the branch label.

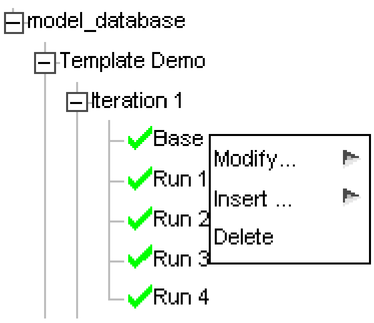
Add Model ... Add a new model into the selected branch.

A menu will be displayed to select a new model and to define the model description that is displayed for the new model.

Insert ... Insert a new branch within the selected branch.

Delete Delete this branch and everything within it.

Right clicking on a model description will display 3 options



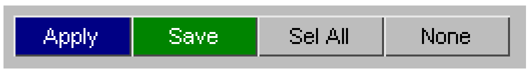
Modify ... Modify the model location and description.

Insert ... Insert a new branch.

The selected model will be moved into the new branch.

Delete Delete the model

Saving the Database



After modifying the database use the **Save** option to save the changes for future sessions.

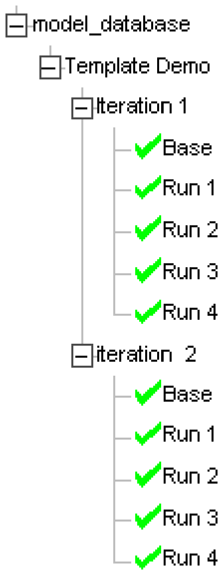
Creating a new Database

If you do not have a database or if you want to create a new one then D3PLOT can create the new database for you. To create a new database click the **CREATE** button and simply enter the name of the new database file in the text box that appears, D3PLOT will then check that the file does not already exist and if it doesn't it will create a new empty database.

Alternatively if you type in the name of a file in the main Open Plot File window that does not exist then D3PLOT will ask if you want to create a new empty database using that filename.

Once you have done this you can use the Modify options above to add items into the database and then save the file before exiting.

Database Format



The Model Database uses an ASCII XML file format.

All items with the database are either branches or models. Each database entry has an XML name and a LABEL element. Models also contain a model element that contains the full pathname of one of the files belonging to the model.

The XML name should be unique and should obey the following rules

- Names can contain letters, numbers, and other characters
- Names must not start with a number or punctuation character
- Names must not start with the letters xml (or XML, or Xml, etc)
- Names cannot contain space

The **LABEL** is the string used to display an item within the tree view. Unlike the XML name the **LABEL** can contain any ASCII character.

Copy Code
XML

```xml
<model_database version="10.000000">
    <Template_Demolabel="Template Demo">
        <iteration_1label="Iteration 1">
            <baselabel="Base"
```

```
                model="e:\release meeting\crush\base\base.ptf"/>
            <run_1 label="Run 1"
                model="e:\release meeting\crush\run1\run1.ptf"/>
            <run_2 label="Run 2"
                model="e:\release meeting\crush\run2\run2.ptf"/>
            <run_3label="Run 3"                model="e:\release
meeting\crush\run3\run3.ptf"/>
            <run_4label="Run 4"
                model="e:\release meeting\crush\run4\run4.ptf"/>
        </iteration_1>
        <iteration_2label="Iteration 2">
            <baselabel="Base"
                model="e:\test\crush2\base\base.ptf"/>
            <run_1 label="Run 1"
                model="e:\test\crush2\run1\run1.ptf"/>
            <run_2label="Run 2"
                model="e:\test\crush2\run2\run2.ptf"/>
            <run_3label="Run 3"
                model="e:\test\crush2\run3\run3.ptf"/>
            <run_4label="Run 4"
model="e:\test\crush2\run4\run4.ptf"/>
        </iteration_2>
    </Template_Demo>
</model_database>
```

7.1.5. File Formats Supported By D3PLOT

File Formats Supported By D3PLOT

A list of the file formats supported by D3PLOT is given in LS-DYNA output files processed and Other output files processed.

Adaptively remeshed analysis filenames

D3PLOT supports LS-DYNA adaptive remeshing, in which a series of families are generated with a mesh that is progressively refined (see Section 4.2.5).

If the "base" **<name>.ptf** (or d3plot) file is selected then all successive families (**.ptfaa** , **.ptfab** , etc) are read in and their states are concatenated internally.

To read in a given family only select is base member (eg **<name>.ptfad**) explicitly and only that remesh family will be read.

Eigenvalue (modal analysis) files

D3PLOT supports output files from modal analyses. They are treated in exactly the same way as transient analyses except that:

- Each "state" is a modeshape, and the States Slider moves between these.

- Animation works on the currently selected state only, oscillating it through +/-180 degrees.

Domain Decomposition files from MPP analyses

The MPP version of LS-DYNA can write a pseudo PTF file that shows the domains into which a model has been decomposed for parallel analysis. It contains undeformed geometry only.

This is not a true PTF file and, in particular, each domain is a part that contains elements of all types in its region. Strictly this is illegal: parts can only contain elements of one type, so D3PLOT handles this automatically as follows:

- The special file type is diagnosed automatically from its contents
- The composite parts of each domain are split into separate solid, beam, shell and thick shell element parts
- These are given labels that make them the same colour in D3PLOT's default colouring scheme
- Groups of each domain, containing all parts in that domain, are automatically constructed.

In this way the technically illegal domain decomposition file can be processed normally, and the use of groups makes the sketching and (un)blanking of domains very straightforward.

7.1.6. Template File

Template File

D3PLOT 10.0 onwards supports a new Template file that can control which windows models are located in and settings for model offsets, colours and initial plotting modes. The contents of the Template file can be applied automatically as models are read in.

All of the options that can be specified in the Edit Window menu (see EDIT WINDOW Changing "Model in Window" attributes) can be set and applied automatically using the template file.

Template File Format

The Template file is a simple ASCII file that controls the position, plotting mode and colour of models with windows. Each line of the template file contains information for a single model/window combination and has the following format.

#
Window=1 model=1 offset=model x=100 y=0 z=0 mode=shaded colour=red

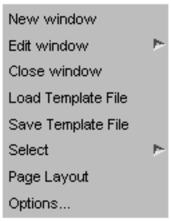
Window=1 model=2 offset=model x=200 y=0 z=0 mode=shaded colour=green
Window=2 model=1 offset=model x=100 y=0 z=0 mode=shaded colour=blue
#

Keyword	Description	Options
Window	Specifies the window number	1-32
model	Specifies the model number	1-32
offset	Offset system. Model - shifts the model in it's own space system Screen - shifts the model in the plane of the screen.	MODEL SCREEN OFF DEFAULT (=MODEL)
x/y/z	Offsets to apply in X,Y and Z directions. By default these are defined in model units. Instead of using model units a %age of the model bounding box can be specified by adding a % to the end of the string x=100 : offset x by 100 model units x=50% : offset x by 50% of the model X dimensions,	
colour	Specify the colour used to display the model. The default option is PART which means that each model is drawn using D3PLOTs normal part colouring scheme where each is drawn in a different colour. Setting this option to a specific colour forces all the parts in the model to be drawn using the specified colour. A user defined colour can also be specified by setting the colour to 0xRRGGBB where RR,GG and BB are the RED, GREEN, and BLUE colour components in the range 0-FF (0-255).	PART RED GREEN BLUE CYAN MAGENTA YELLOW RED_MAGENTA LIGHT_ORANGE YELLOW_GREEN GREEN_CYAN CYAN_BLUE DARK_ORANGE LIGHT_BLUE GREY BLACK WHITE DEFAULT (=PART)

mode	Specifies the default drawing mode for the model	SHADED WIRE HIDDEN CURRENT DEFAULT (=CURRENT)

Loading a Template File

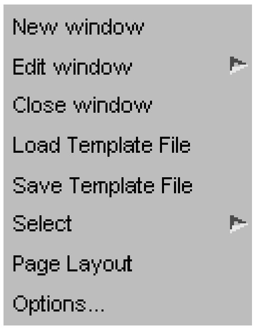
If a template file is specified in the Open Model panel it's contents will be applied automatically as model are open and read into D3PLOT.



As well as applying template settings automatically as models are read in a template file can also be loaded at anytime via the **WINDOW>** popup menu. As the file is read the settings it contains are applied to any Window/model combinaitons taht match those defined in the file

Shortcut keys can also be assigned to read and apply template files (see Shortcut Keys for more details)

Saving a Template File

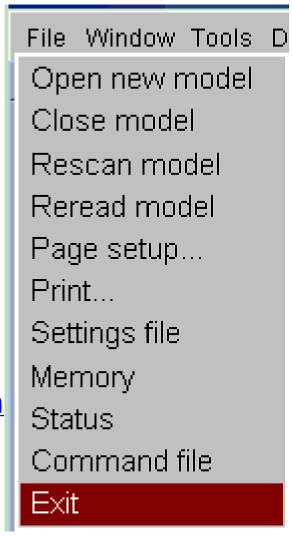


The current window and model layout information can be saved into a template file at anytime via the **WINDOW>** popup menu.

When a template file is saved any offsets that have been specified will be written out using the same format as the option in the Edit Window menu (see OFFSET... adding an artifical offset to models in windows for more details). If for example modle offsets are being displayed as %ages in the Edit Window menu then all the offsets will be converted to %ages before they are written to the templaet file.

7.1.7. Open New Model Opening a New Model File

Open New Model Opening a new model file



From D3PLOT 9.3 onwards, D3PLOT supports up to 32 concurrent models.

You can open a new model file at any time using the **FILE>** popup which is located at the top left of the top options (**Main Menu**) box, **OPEN NEW MODEL** command.

This creates a new graphics window to contain the model, then reads a new filename in exactly the same way as described in Open a Single Model above.

If this is a second or subsequent model its window will inherit those attributes of the first window that can legitimately be transferred to this new one: background colour, data component, cut-sections, etc. Where attributes cannot be transferred, for example a data component that doesn't exist in the new model, then the programme defaults will be used.

Close Model Closing an existing model

You can close a model at any time using this command. The model will be removed from any window in which it appears, and if it was the only model in such a window then the window will also be deleted and any remaining windows renumbered downwards to close the window numbering gap.

(Note that in D3PLOT V92 models may reside in the database without being displayed in any window, however they will continue to consume memory and models should be deleted if unneeded to free memory for other purposes.)

Rescan Model Scanning a running analysis for more states

If an analysis is still running you can scan the file family for any further states that may have been written since you opened the model (or last scanned it).

Because of the way that computers work it is possible that the most recent state written from a running analysis may not be completely debuffered to disk. It this case D3PLOT will usually detect that the state is incomplete and offer you options for dealing with this, however it is usually best to ignore such states as attempting to read corrupt or incomplete data can lead to problems.

Reread Model Closing and reopening a model

The **Reread Model** command differs from **Rescan** in that it closes a model completely then reopens it again (in the same window(s) that it occupied previously). It is the equivalent of **Close Model** followed by **Open New Model** .

You should use this instead of a **Rescan** when:

- An analysis file that is currently open in D3PLOT has been rerun from scratch using the same filenames, and needs to be reread in its entirety.
- You have an adaptive remesh analysis running and you want to scan for further family members (**Rescan** will only look within the current family member).

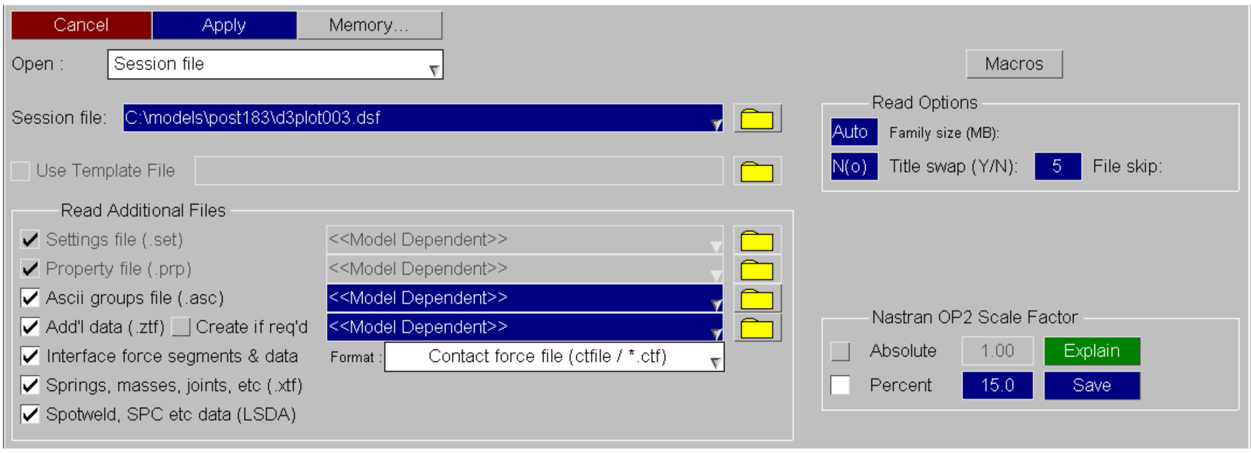
7.1.8. Session File

Session File

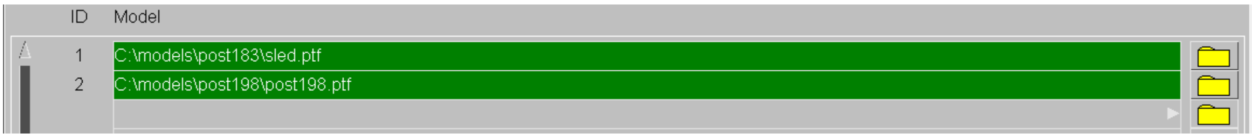
From D3PLOT 16.0 onwards, a D3PLOT session file may be saved and later restored, as is, to return users to the previous state. The session file contains information including results filenames, window and page layout, and model properties.

If the session file was saved while the D3PLOT->T/HIS link (see T/HIS the D3PLOT <=> T/HIS link) was running then the session file will also contain information of the number of T/HIS graphs and it will contain a T/HIS FAST-TCF script which will regenerate the graphs contents.

This option can be used to reload a version 16 session file and to restore previously loaded models, window layouts and model-specific information.



Similar sets of results corresponding to different runs may be chosen using the model selection part of the menu.



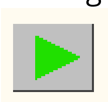
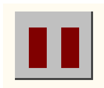
7.2. Basic Animation the "Current State" and Selecting States

Basic animation, the "current state", and selecting states.

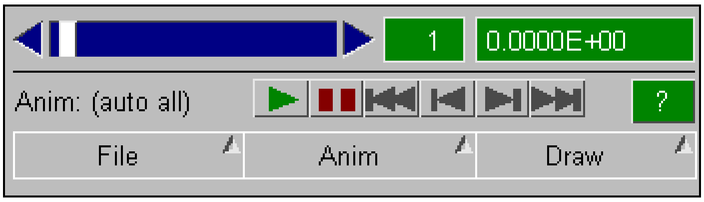
The programme maintains the concept of a "current state" for each window, which is that being displayed at the moment. When an animation is halted the state at which it stops becomes the current state, and any state selected explicitly by the user subsequently becomes the current state in its place.

Each graphics window is independent, and each may show different states. To provide both independent and collective control the following mechanisms are used:

The master **STATE NUMBER** slider applies to ***all*** windows for which its **W *n*** tabs are active.

Moving this slider to a new state will cause all these windows to jump to this state. **PLAY**  makes all selected windows animate, and **STOP**  halts them.

This range of this slider is 0 to the highest state in all models, and it can be the case that it permits selection of a state that doesn't exist in a given model. Selecting such a state is legal and leaves the window(s) of that model unchanged.

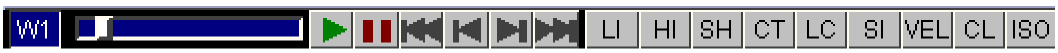


The local state slider and associated controls in the button bar at the top of each graphics window applies to this window ***only*** .

Moving the slider, or using **<<, |<** , etc, lets you move between the animation states currently defined for this window.

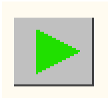
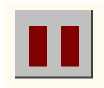
By default all states in a model are selected for animation; but this can be limited to restricted states, or extra states can be displayed by interpolation.

Therefore these controls move between what has been selected for this window (or for modal analyses through the +/- 180 degree phase angle for the current modeshape).



All animation and static state selection is carried out in the **State Display** box. Its basic controls are described here, with more detail in the following sections.

To start animating

To initiate an animation simply press **PLAY**  , and to halt it press **STOP**  .

By default all states in the model will be animated at full speed in the current display mode. More information on animation is given in Animation How to display, control, store and retrieve animation sequences .

To select an explicit state by number

Either: Move the **STATE NUMBER** slider to the state you want, or use the arrows at its ends to scroll it left or right. When you release the mouse button the selected state becomes current and will be drawn.

Or: Type an explicit state number into the **State:** box. This will become the current state and will be drawn.

To select an explicit state by time

Type the required time into the **Time:** box. If a state with that time exists it will be used, otherwise the state with the closest time after that you have specified is used. This becomes the current state and is drawn.

The programme "knows" about all states in a file, they are scanned as part of the initialisation process, and it can jump directly to any state. Data required for plotting are read in selectively on as "as needed" basis.

States at interpolated times

The current (static) state shown in this box cannot be at an interpolated time between two explicit states: defining an intermediate time will result in the next highest state being used. However the sliders and state manipulation controls in each window can be used to move between interpolated states.

For more information on interpolation see SET_STATES > Selecting the States to Be Animated**.**

Selecting and animating mode-shapes.

Most analyses are transient, and each state will show successive times. Modal (eigenvalue) analyses are performed in the frequency domain, and each "state" is a different modeshape at some frequency.

D3PLOT operates in much the same way except that:

- The "States" slider now moves between modeshapes (state 1 = mode 1, etc)
- Animation is performed by oscillating a single state through +/- 180 degrees
- The slider at the top of each window moves through the +/- 180 deg cycle.

Using keyboard "short cut" keys to cycle through states

You can use the following keyboard keys to select states:

<-- and --> arrow keys step backwards and forwards respectively. They loop round when they reach the limits of their respective directions.

<Home> jumps to the first state
<End> jumps to the last state

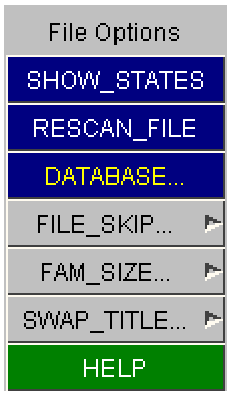
As with all short cut keys the windows upon which they act are determined as follows:

- If the mouse is in a graphics window then they act only upon that window.
- If the mouse is in some other menu window then they act upon all active graphics windows.

To cycle through animation ***frames*** , as opposed to states, use <shift> + left/right arrow keys. In most cases the only difference will be in data-bearing plots with contour levels set to "auto":

- Cycling through states will autoscale each image.
- Cycling through frames will use the envelope of contour values for the whole animation.

The **FILE >** popup menu options



You can manage many aspects of your database using the **FILE >** popup menu.

Some of these options can only apply to one model at a time. When this is the case the operation will be applied to the first active model as selected by the **W *n*** tabs in this panel.

SHOW_STATES - List all states in the file
If you think that you cannot see all the states that should be there you should consider the following possible reasons:

- Analysis is still running: You may need to **RESCAN_FILE** for newly created states.
- Missing family members: You may need to adjust the **FILE_SKIP** value.
- Wrong family member size: You may need to adjust the **FAM_SIZE** value.

RESCAN_FILE - Scan the file for any new states.

If your analysis is still running D3PLOT will not know about any states that may have been written since its initial scan of the file. **RESCAN_FILE** will search the file for new states and update the internal tables to show them.

DATABASE - Displaying and managing the results storage database D3PLOT loads results from file on as "as needed" basis, and may supersede unwanted results in memory to save space. The process is automatic and can normally be ignored, however users with big models may need to intervene to economise on memory usage.

FILE_SKIP - Jumping over gaps in family member sequence

Sometimes the family member sequence **`<name>.ptf`** , **`<name>.ptf01`** , ... **`<name>.ptfnn`** may contain gaps. This can be due to deliberate deletion of intermediate members to save disk space, or because LS-DYNA has skipped a member.

D3PLOT will skip over **< FILE_SKIP >** gaps before giving up its search for new members and deciding that it has reached the end of the file family. **< FILE_SKIP >** may be zero (no gaps) or any positive integer, but bear in mind that large values will slow down disk scanning as many non-existent files are searched for. If you change this value the family will be re-scanned automatically to detect any new members this may have made visible.

FAM_SIZE - Setting the file family size to use.

The file family size from LS-DYNA defaults to 7MBytes, but is sometimes set to some other value (using the X= parameter on the input line). D3PLOT can determine the member size of each family automatically by taking the larger of the first two members rounded up to the nearest Mbyte.

However you can override this value if, for some reason, the automatic method does not give the correct answer. Doing so causes the file family to be re-scanned automatically to detect any changes. Setting the value to zero effectively returns it to "automatic" mode.

SWAP_TITLE - Unscrambling endian-swapped titlesIf the analysis title appears to have every 4 letters reversed (ie ABCDEFGH = DCBAHGFE) then it has probably been (incorrectly) endian-swapped by your version of LS-DYNA. You can correct this by swapping between reversed and normal modes.

Setting these parameters externally

The parameters on this page may be set externally (or in the Shell) with the following environment variables:

```
setenv FILE_SKIP 10
FILE_SKIP=5, export FILE_SKIP
```

(Unix, C shell syntax)

(Unix, Bourne/Korn shell syntax)

```
setenv FAM_SIZE 7
FAM_SIZE=0; export FAM_SIZE
```

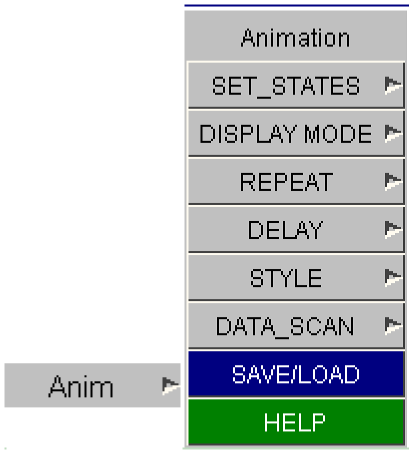
```
setenv SWAP_LSTC_TITLE true
SWAP_LSTC_TITLE=false; export SWAP_LSTC_TITLE
```

Windows users may set these variables in the **System** , **Environment** panel.

The Animation options popup menu

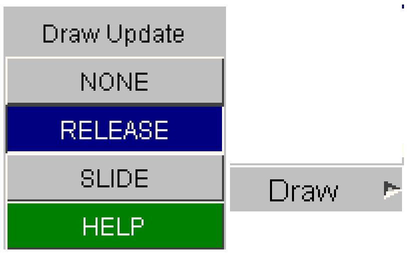
The **ANIM >** popup menu provides access to the options which control the extent, speed, and many other attributes of animation.

These, and many other aspects of animation, are described in Animation How to Display, Control, Store and Retrieve Animation Sequences.



The drawing vs. state selection menu.

The **DRAW >** popup menu controls how often (and if) the current state gets drawn when a new state is selected using the **STATE NUMBER** slider.



NONE	The newly selected current state is not redrawn at all.
RELEASE	The new state is drawn when you release the slider.
SLIDE	As you move the slider between states each one you pass gets drawn.

The default is **RELEASE** since this reduces drawing time for large models when selecting states.

Note that **SLIDE** can be used to scroll through (visually) an animation, but the same effect is achieved more easily using the frame slider bar on top of the graphics window: see Animation How to Display, Control, Store and Retrieve Animation Sequences .

The meaning of fake state #0

You may have noticed in some of the examples above that there is a state #0. This is a special state assembled within D3PLOT from the undeformed geometry with all displacements, stresses, etc set to zero. It is given the time 0.0. It exists for the following reasons:

- "Real" complete states in database files normally range from #1 to #n; but it is possible to create a database that only contains geometry, and has no complete states. Since all plotting commands within D3PLOT require a current state to be present a file with no states needs the "fake" state #0 be synthesised to permit plotting.
- State #1 may have the time 0.0, but may not contain the "undeformed" configuration: for example if an analysis started with a dynamic relaxation or from pre-defined displacements. In this situation reading in "fake" state #0 allows you to plot the undeformed configuration in all modes.

State #0 is ignored for purposes other than static plotting: it cannot be included in an animation, or used for time-history output

Support for analyses using "Adaptive Remeshing"

The adaptive remeshing facility in LS-DYNA, used primarily for metal-forming, generates output files in a sequence that is different to those from "normal" analyses.

Each remesh effectively constitutes a new analysis in which the quantity (and labelling) of nodes and elements in the remeshed parts will have changed. A new file family is generated at each remesh, and LS-DYNA flags these by appending "aa", "ab", etc to the output filenames for the 1st, 2nd, and so on families.

D3PLOT is able to detect this, so long as you give the name of the original analysis as the input file, and will automatically scan all the file families extracting their times, so that remeshed states are detected and made available. Thereafter you may use the programme in the usual fashion: selecting a new state automatically uses results from the correct file family, and capabilities such as animation work in the normal way.

There are a few minor limitations:

- Since each file family is a new analysis (it has different topology and geometry) D3PLOT has to maintain separate internal tables for each family. This requires extra memory, and you will find that overall memory consumption for adaptive analyses is greater than for a "normal" analysis of equivalent size.
  If you run out of memory trying to process all the remesh files together you may need to open its families individually. (See the MEMORY button for more information about viewing and modified database memory usage.)
- Blanking, colour, transparency, and related "property" changes are propagated through all families in an analysis sequence. However the situation can arise that an element in analysis A does not exist in analysis B. This is not an error, but it may lead to patchy looking plots. Therefore when changing visual

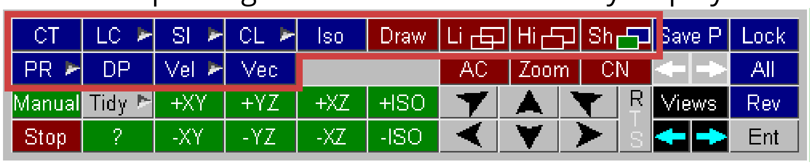
properties of remeshed parts it is best to operate at the **PART** level (which will exist in all families), rather than on individual elements (which may not). Elements of parts that are not remeshed do not suffer from this restriction.

- Operations in **XY_PLOT** which imply collecting data across families will not work. It is not possible to extract "time-history" data in a consistent fashion across remesh (ie file family) boundaries.
- Using time interpolation for animation will only work within a remesh family: it is not possible to interpolate across remesh boundaries since the process requires nodal and element values to be interpolated between adjacent states. If you attempt to do this you will get a warning message, and the animation will "jump" to the first state in the new family.
- Reference geometry will not work in conjunction with adaptive remeshing, for the same reasons that interpolation will not: reference between two incompatible states is invalid

7.3. Displaying Geometry and Results

Displaying geometry and results.

D3PLOT contains a number of different plotting mode. Some of these only display geometry while others display data values.



7.3.1. Drawing Commands that Do Not Plot Data

Drawing commands that do not plot data

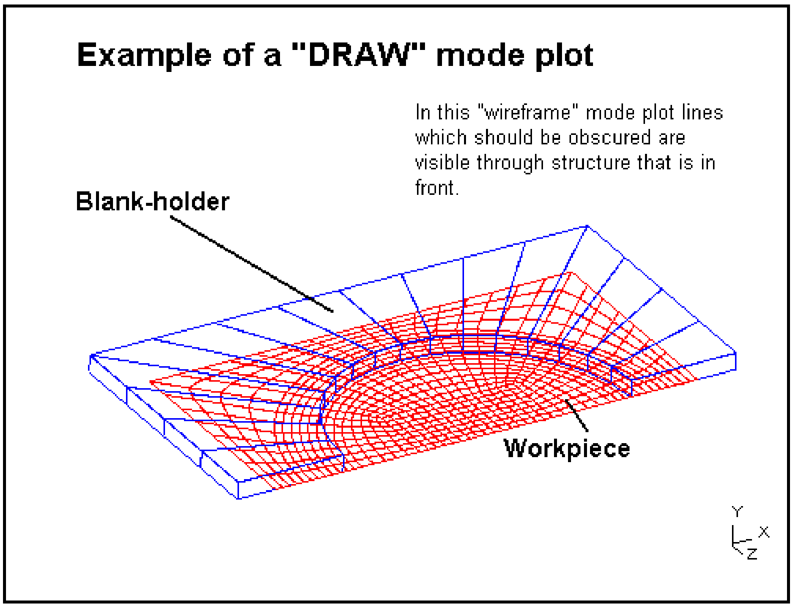
These commands are always available since they do not display any data.

DR (alias DRAW)

Draws the undeformed geometry in wire-frame mode, the current in-core state is irrelevant.

No time is shown since none is associated with the undeformed geometry.

"Wire-frame" displays do not obscure hidden lines.

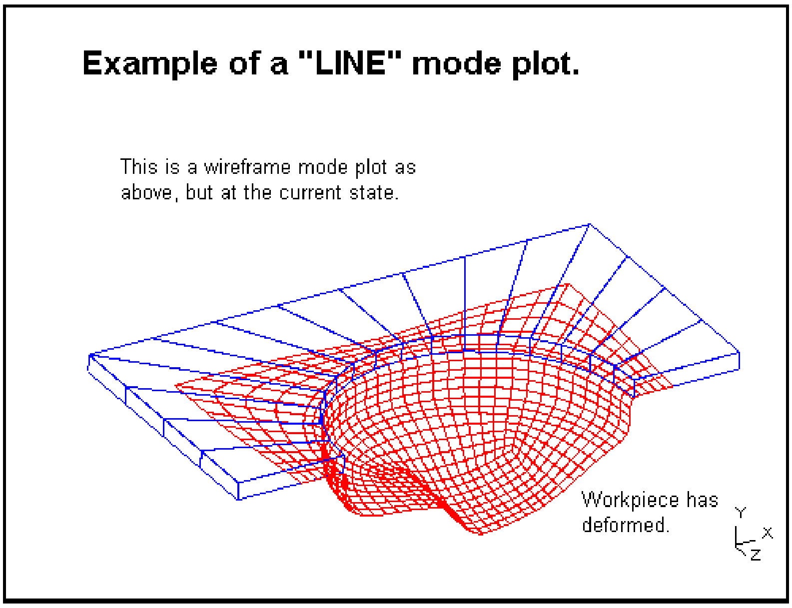


LI (alias LINE)

Draws the current in-core state in wire-frame mode.

A time is now shown (in this case 4.6ms) since this represents data at that time.

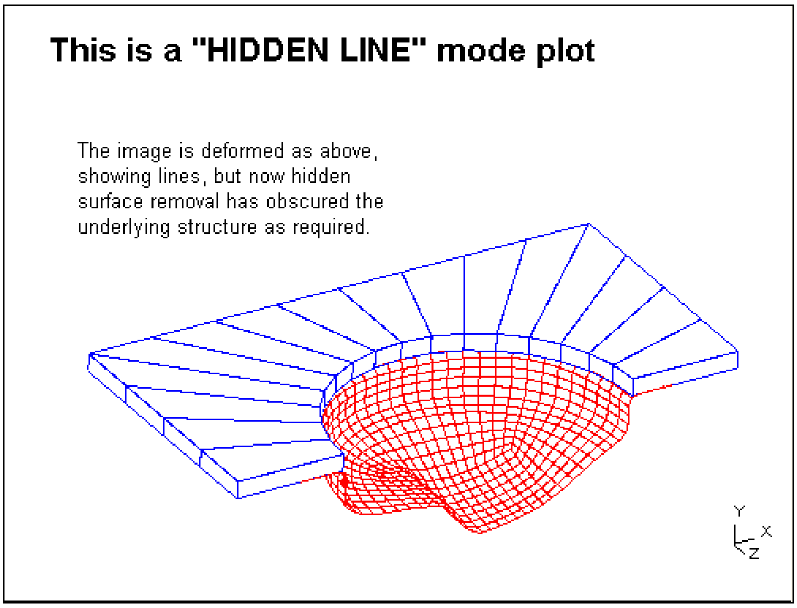
Wire-frame mode still exposes lines which should be hidden.



HI (alias HIDDEN_LINE)

Draws the current in-core state in hidden surface mode.

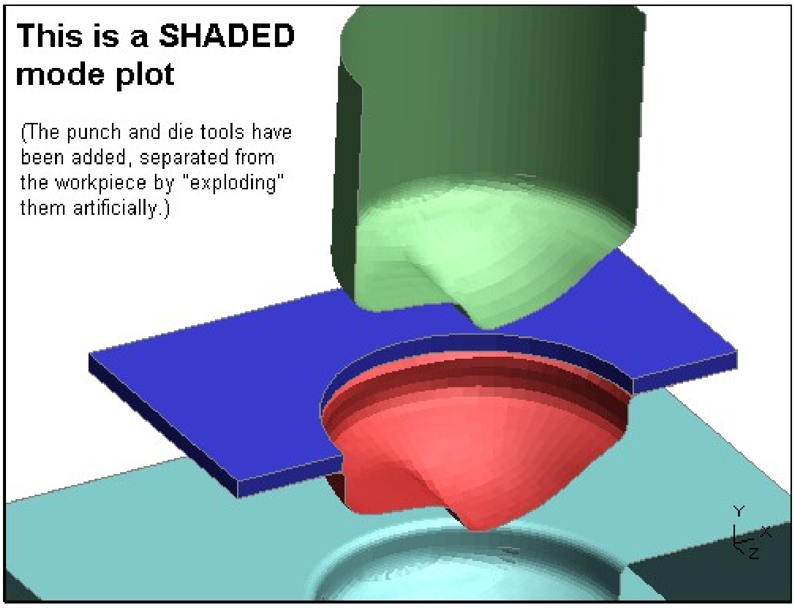
This plot shows the deformed shape at 4.6ms as before, but now hidden lines have been removed .



SH (alias SHADED)

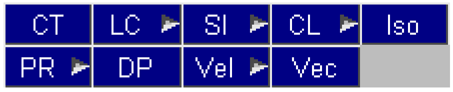
Draws the current in-core state in lit and shaded "greyscale" mode, implicitly with hidden surfaces removed. The command-line equivalent command is `[Greyscale] GO`.

Shading has been applied, assuming a light source at the observer, and hidden-surface mode is implicit: any hidden lines will be removed.



7.3.2. Drawing Commands that Plot Data

Drawing commands that plot data

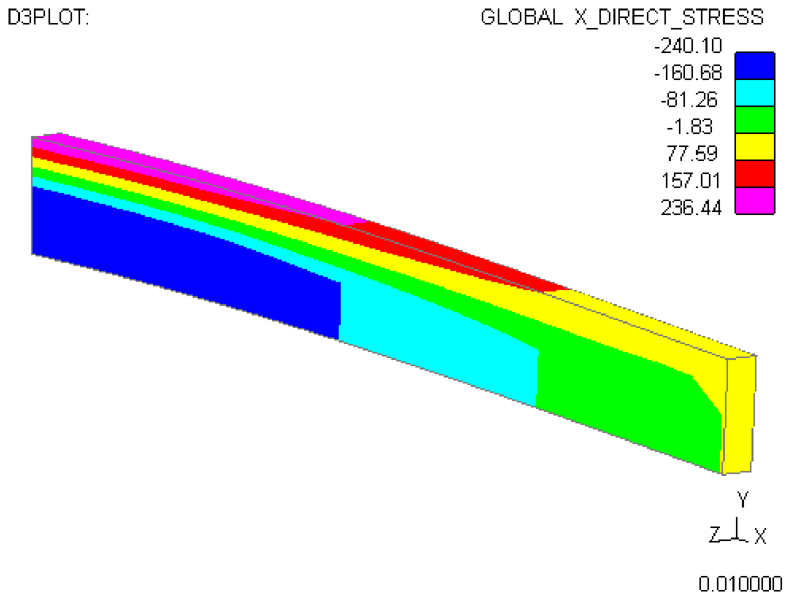


The plotting modes available for plotting data depend on the currently selected data component (see [Selecting Data Components](#))

7.3.2.1. CT

CT

A continuous-tone plot draws bands of contours in solid colours. No account is taken of lighting.



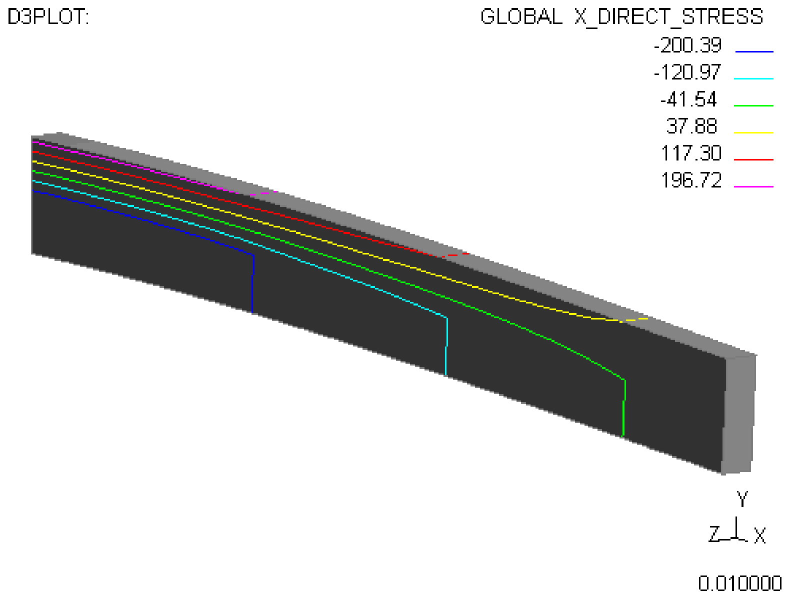
7.3.2.2. LC

LC

A line-contour plot draws lines of constant value across elements.

Note that the contour values, and hence the lines, lie in the middle of the equivalent solid bands above.

By default LC mode plots are drawn as lines on a shaded element mesh, but "mixed-mode" plotting allows them to be superimposed on hidden or wireframe meshes too. See the PROPS box in [Properties: Controlling colour, drawing style, transparency, lighting attributes and overlay of entities](#)

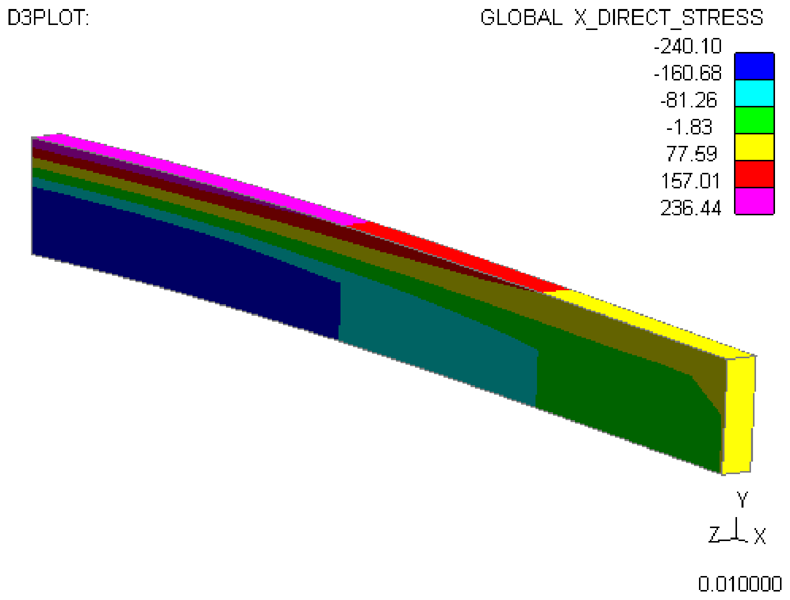


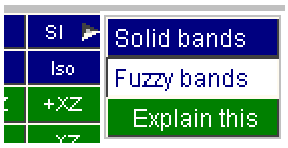
7.3.2.3. SI

SI

This is similar to a continuous-tone plot, but lighting is included to give more impression of shape.

The **LIGHT** command, see The LIGHT panel, controls the lighting aspects of the plot; and there are special options in **CONTOUR**, see DATA COMPONENTS - BASIC, to control the contouring aspects.





"Fuzzy" vs "Solid banded" plots in SI Shaded Image mode.

Prior to D3PLOT 9.2 "SI" mode plots always had blurred contour bands, making it impossible (deliberately) to tell where band boundaries lay.

From D3PLOT 9.2 onwards the default is now for SI mode to show solid contours, giving the effect of a lit CT mode plot, although you have the option of reverting to "fuzzy" (technically "Gouraud") shading if you wish.

There are considerable speed differences when rendering:

Solid Bands (Default)	Draws solid contour bands, exactly as in a **CT** plot, except that lighting is added. Execute more slowly since discrete polygons of colour must be computed for elements with data variation, often doubling or more the quantity of data sent down the graphics pipeline.
Fuzzy bands	Execute faster as 3D graphics hardware is optimised to handle gouraud shading efficiently. Also "smooth" shading is not implemented with Banded SI plots, as the overhead of interpolating outward normal vectors at sub-polygon vertices would be prohibitive.

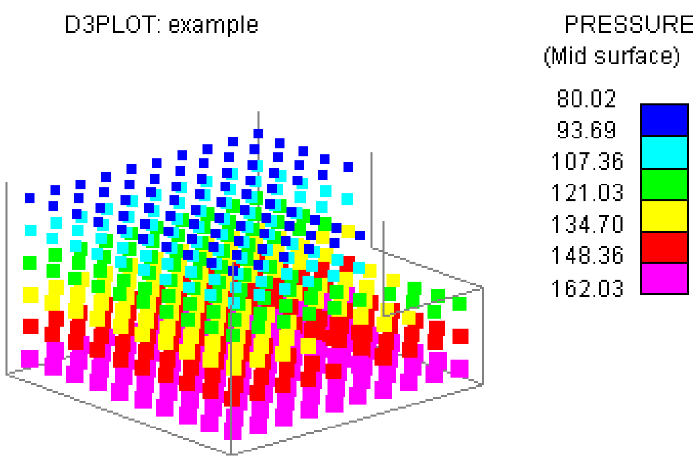
7.3.2.4. CL

CL

A cloud plot produces points displaying the value of the selected data component at each node. These can been drawn as a fixed size of the user's choice or proportionally to their value. This property can be set with the **CONTOUR** command - see DATA COMPONENTS - BASIC .

CL plots are a far "cheaper" way of seeing what is going on inside large solid meshes than ISO plots (or cut sections). This is because they don't have to worry about data averaging or complicated graphics, making them much faster to process.

In addition drawing "blobs" is fast in hardware, making this a quick method of displaying data from very large models.



Two data computation options are provided for element-derived data displayed in **CL** oud plots:

Element centre (default)	Raw (unaveraged) element values are displayed at the element centre locations
Averaged at nodes	Element values are averaged at nodes, and displayed at the node locations

In the nodally averaged case the results shown are equivalent to those seen in low resolution contouring, which has the effect of "smearing out" the peak and trough values at element centres.

Nodally-derived data is always plotted unconditionally at nodal locations and, by definition, is not averaged in any way.

7.3.2.5. ISO

ISO

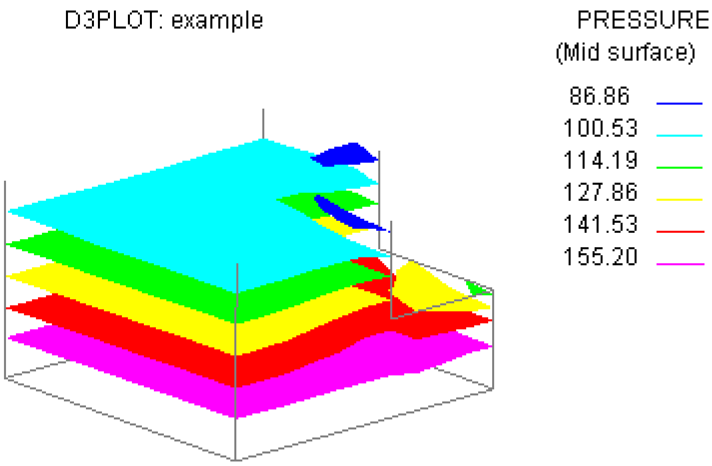
ISO surface plots draws surfaces at constant values, with lighting added.

In 3D elements, as shown here, the contours will show surfaces of constant value through the body of the solid mass.

In 2D elements the effect is the same as a Line Contour plot, drawing lines of constant value.

Note that ISO plots of large solid meshes can be many times slower than, for example, SI plots of the outer surface. This is due to the "cube /square" law: an ISO plot has to consider all elements inside the mesh, whereas an SI plot need only consider external elements.

Note also that ISO plots are not supported for solid elements with multiple integration point results in high resolution and all integration points. If high resolution is selected for such elements the integration point will be automatically changed to average at centre.



7.3.2.6. PR

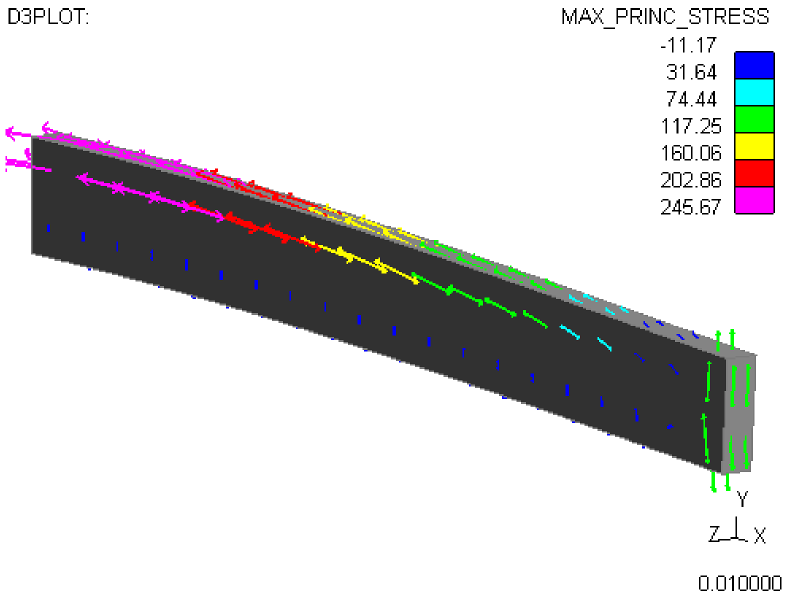
PR

A Principal Stress/Strain plot displays values as vectors. These vectors can only be displayed on 2D and 3D elements (solids and shells).

This option will only be available if the current data component is either a principal stress or principal strain component.

By default PR plots are drawn as lines on a shaded element mesh, but "mixed-mode" plotting allows them to be superimposed on hidden or wireframe meshes too. See the PROPS box in Properties: Controlling colour, drawing style, transparency, lighting attributes and overlay of entities

Note that for solid elements principal stress/strain vectors are always shown averaged at the centre, even for solid elements with results at multiple integration points.



7.3.2.7. DP

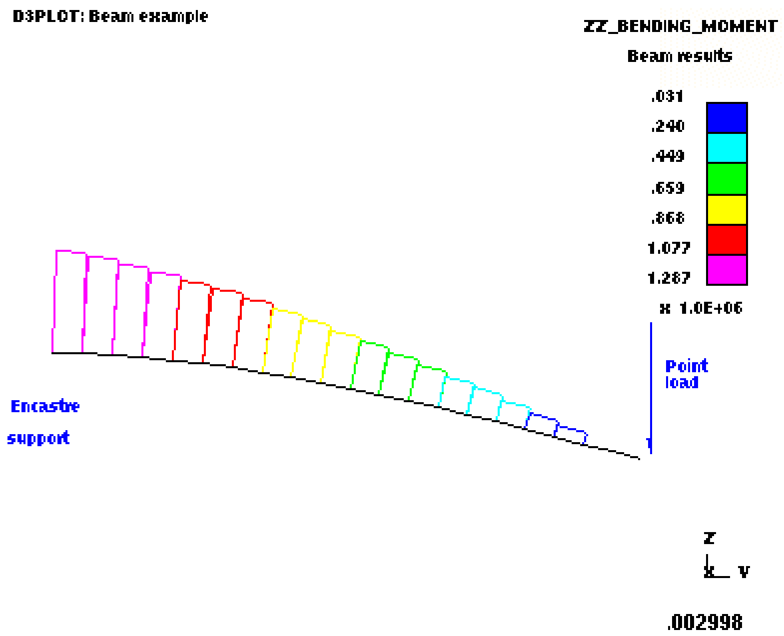
DP

This option will only be available for data components that apply to beam elements.

This shows the **DP** option: a "diagram" plot showing results hatched on the beam.

Hatching size is proportional to data magnitude, and colour also follows the normal contour band limits. Directional data is displayed by default on the relevant local beam axis.

Any data component can be displayed this way, not just bending moments.

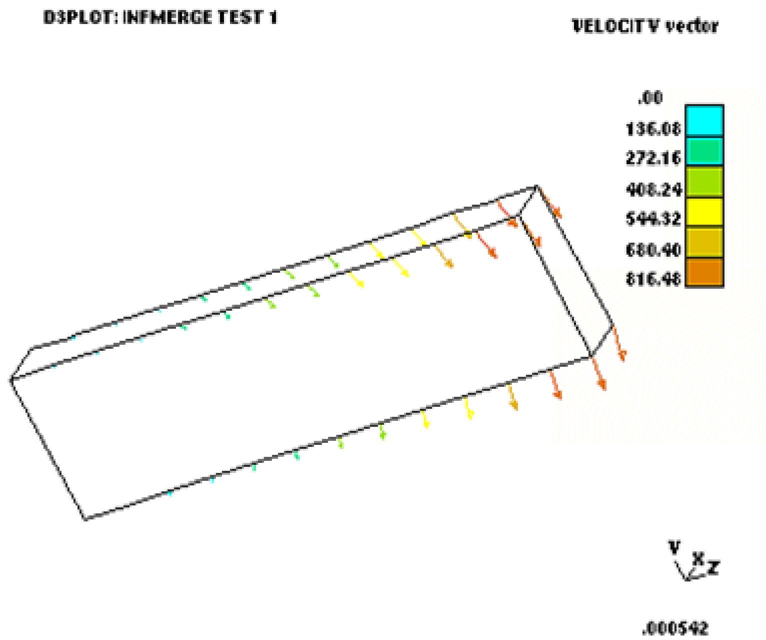


7.3.2.8. VEL

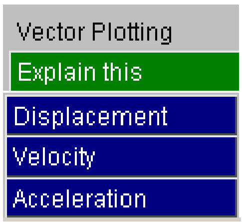
VEL

A velocity plot draws arrows showing the direction and magnitude of nodal velocities.

As plotting nodal velocities is something that is very commonly used the VEL button ignores the currently selected data component and temporarily swaps the component to velocity vector.



The popup menu attached to the VEL button can be used to easily swap the data component to either Displacement or Acceleration.



The arrow length is proportional to velocity magnitude, and arrow colour is also based on magnitude using the normal contour band colours and bands. Both of these attributes can be modified if required - for more details see Vec Plots

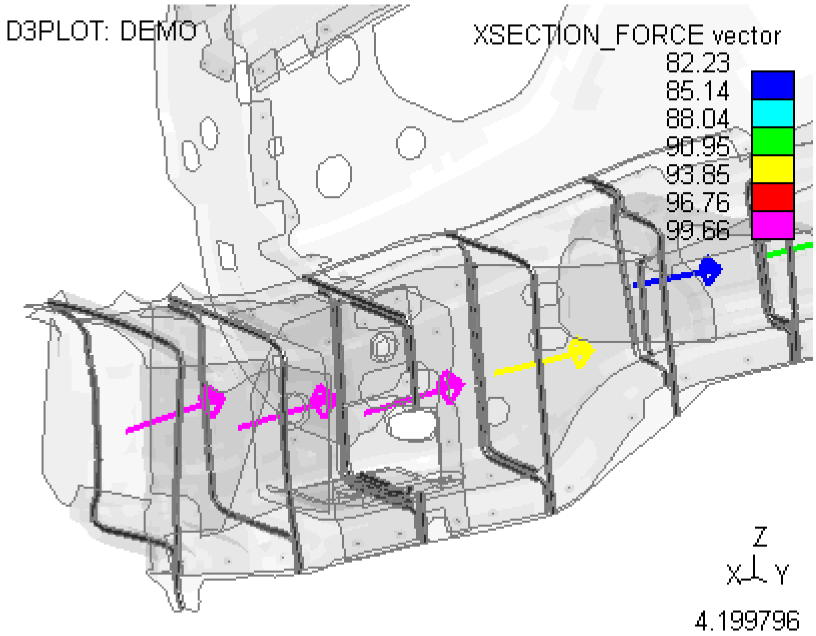
Like LC and PR plots **VEL** ocity plots are drawn on a hidden-line mesh by default. However mixed-mode plotting means that the underlying mesh may be rendered in shaded and/or wireframe modes - see the **PROPS** box in Properties: Controlling Colour, Drawing Style, Transparency, Lighting Attributes and Overlay of Entities .

7.3.2.9. VEC

VEC

A **VEC** tor plot is similar to a velocity plot except that it uses the currently selected data component.

This option will only be available if the current data component is a vector quantity.



7.3.3. Visual Controls

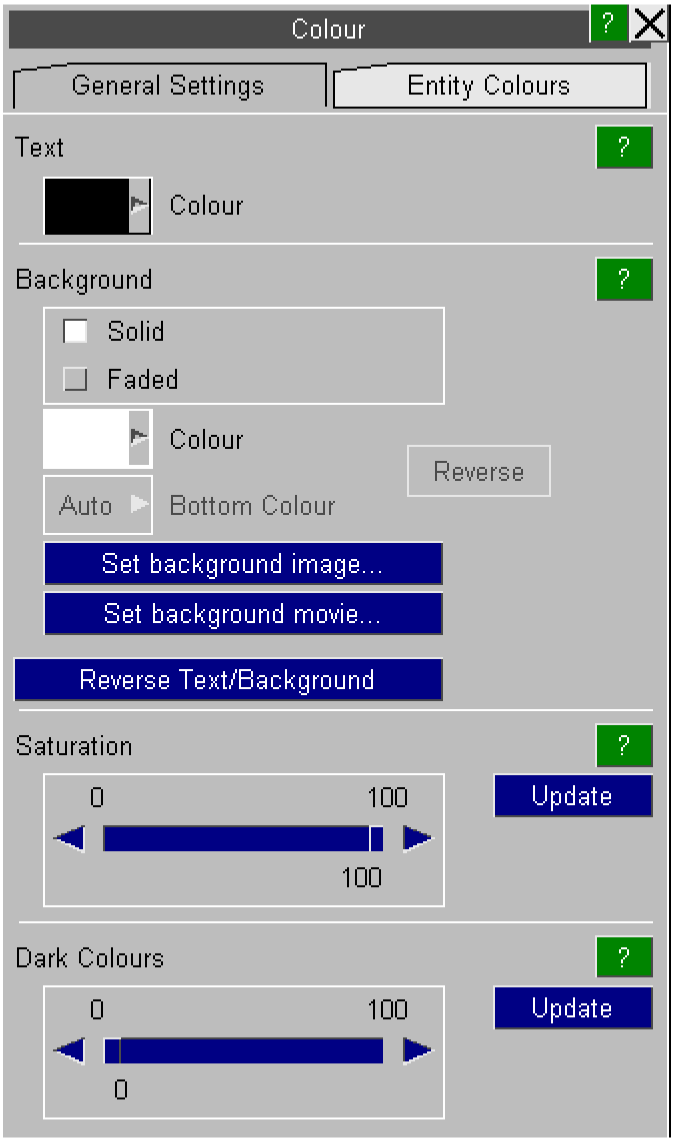
7.3.3.1. COLOUR: Controlling the Colours Used for the Display

COLOUR: Controlling the colours used for the display

Note that changing element and part colours can also be achieved, usually more easily, using:

- The **Properties** panel, where all visual properties can be changed, saved and restored.
- " Quick Pick " screen selection, which is by far the quickest method.

The **COLOUR** command controls general colour usage in D3PLOT except the assignment of contour colours (see Changing contour band colours).



7.3.3.1.1. "General Settings": Commands that Affect All Plotting Modes Are

"General settings" : commands that affect all plotting modes are

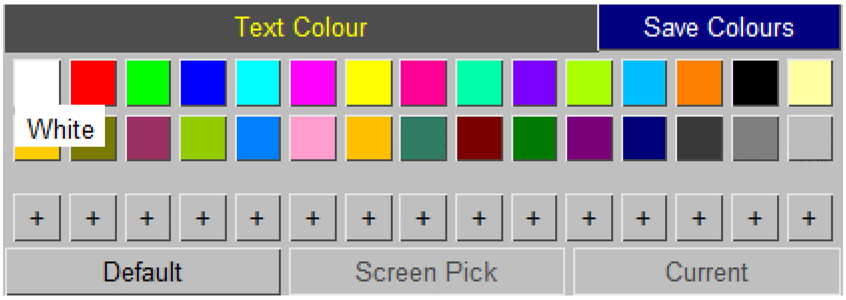
Text

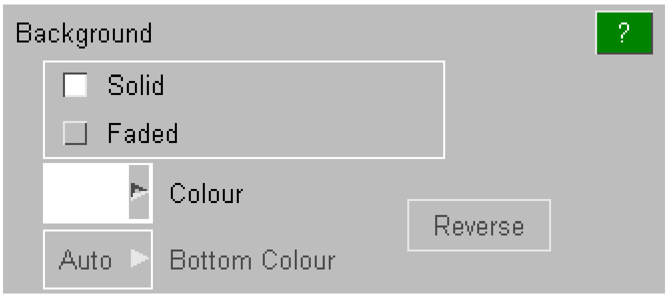
Controls the colour used for text (ie clock, header, etc). By default this is white but, as with the background, you can make this a standard colour or a user-defined shade.

Text and background colours may be chosen from one of the 30 standard colours shown here. Alternatively you can use **DEFAULT** which resets the relevant colour to its default. That is white for text, and black for the screen background.

Background

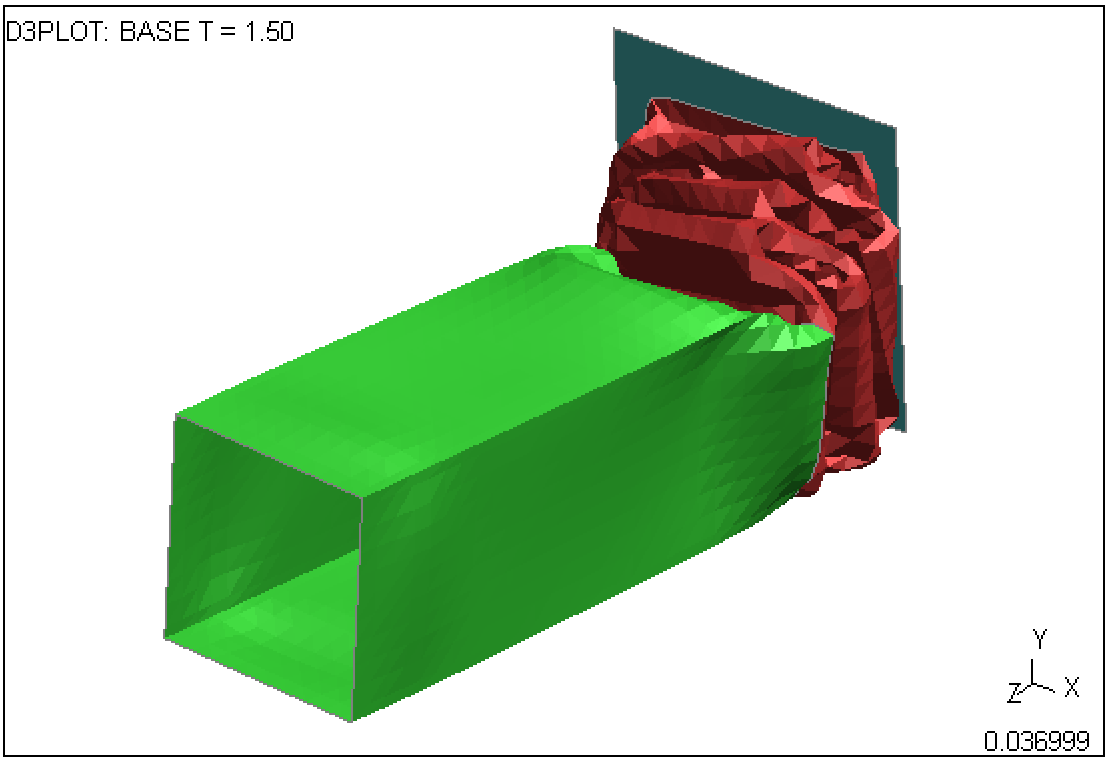
Sets the graphics window background colour. By default this is black, but you can choose from a range of standard colours, or make your own user-defined shade.





Solid Background

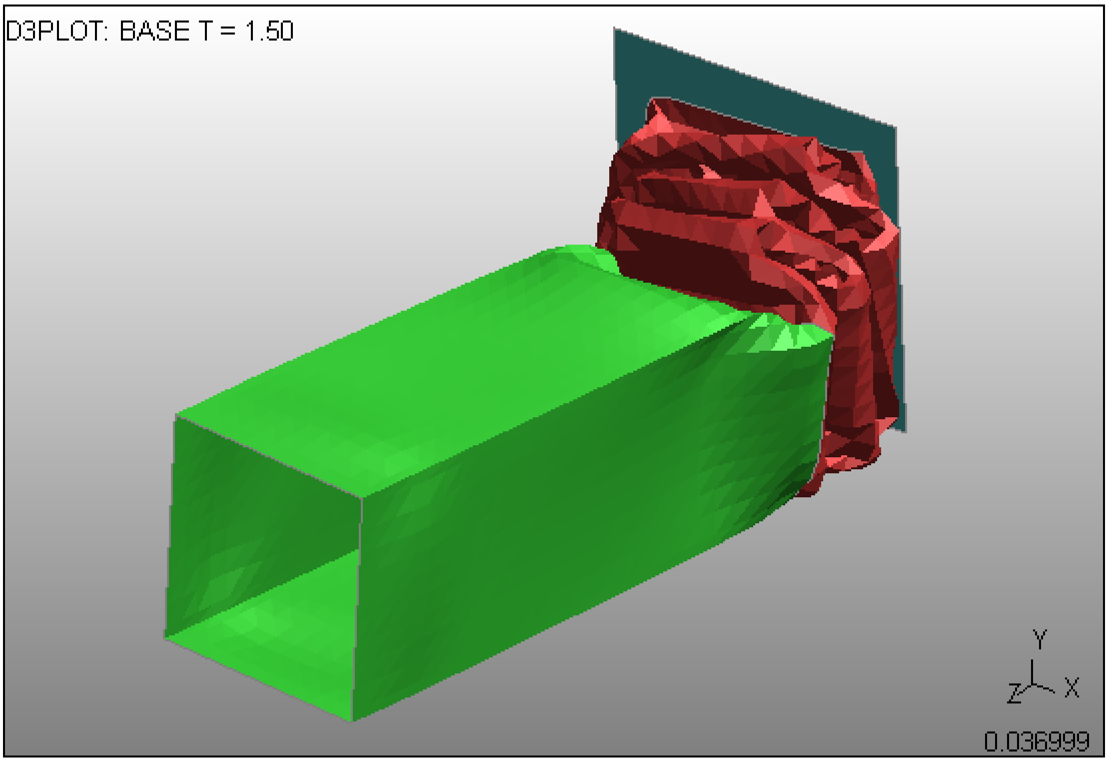
By default the background is drawn using a solid colour.



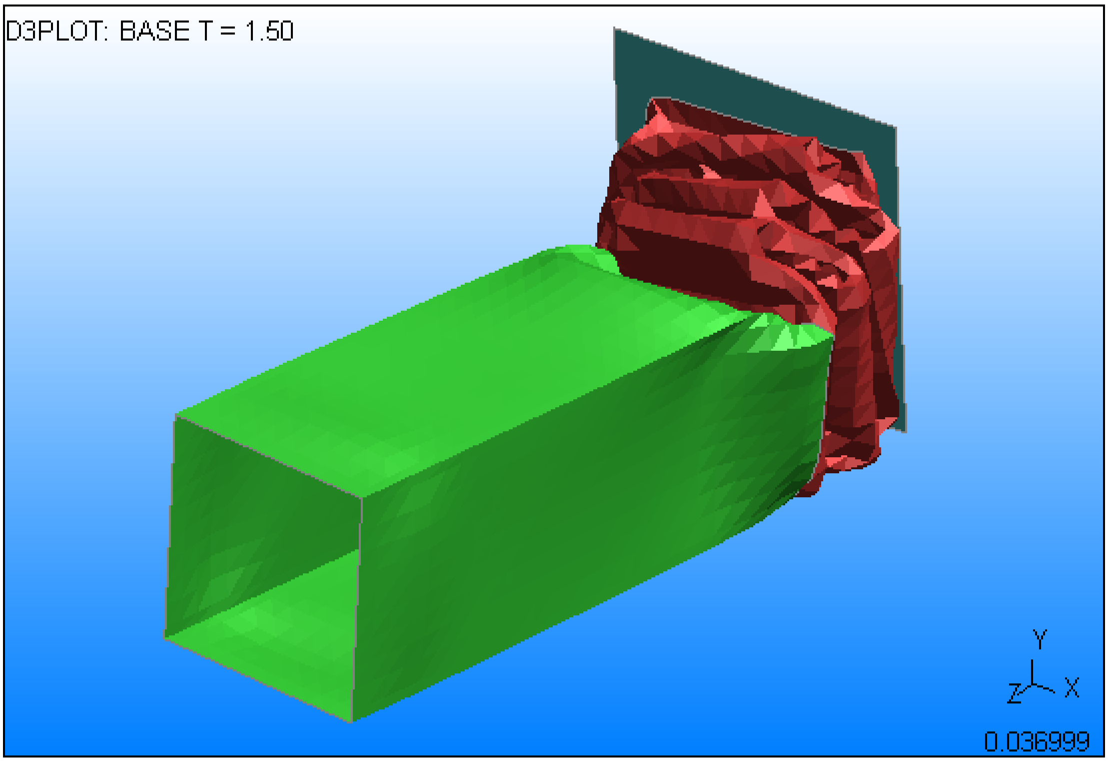
Faded Background

The Faded option draws a background which fades from one colour to another as you move down the screen.

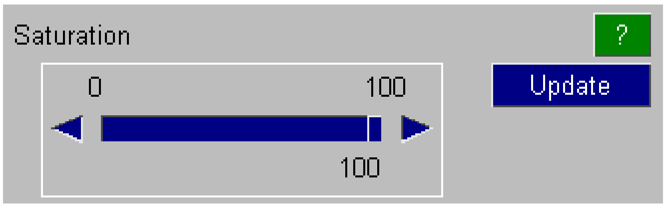
By default the top of the screen is set to the defined background colour value and the bottom of the screen is set to a colour with 50% of the RGB values of the background colour.



Instead of using this automatic colour a separate background colour can also be defined.

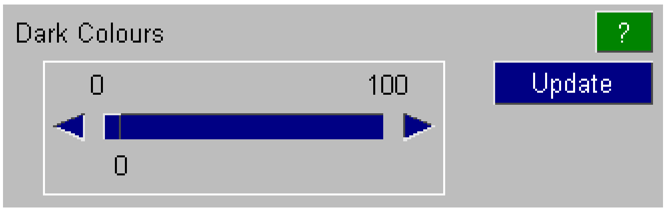


Saturation



Controls the saturation of all colours in the range 0% (grey) to 100% (fully saturated). Normally all colours are fully saturated (ie 100%) to give the brightest possible display, but you may need to desaturate colours in some circumstances, for example when capturing frames for a video.

Dark Colours

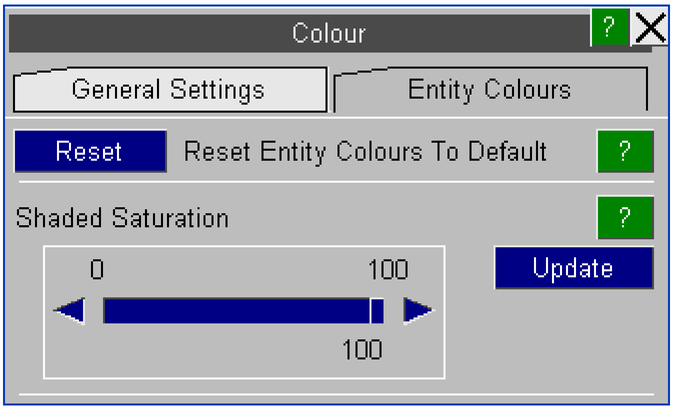


Is used to lighten colours for colour printers. Most colour printers use cyan, magenta, yellow and black inks and they tend to render darker colours such as blue and magenta too darkly.

This command lightens these colours preferentially so that colour plots look better, even though the display may look strange. Its default value is 0% (ie no lightening), and it may range to 100% (which will turn blues into white). You will need to experiment with your plotter to find the best value, a suggested starting point is 50%.

7.3.3.1.2. "Entity Settings": Commands that Affect Entity Colours

"Entity settings" : commands that affect entity colours



Reset

Resets all entity colours to their standard D3PLOT defaults.

Shaded Saturation

This has the effect of desaturating "shaded" images, which gives them a more pleasant appearance. It has no effect on data plotting modes (e.g. **SI**, **CT**) or on wireframe/hidden element borders.

Note also that the "global" **SATURATION** control remains. It may be used to desaturate all images. The effects of the "global" and "shaded" saturation controls are additive.

7.3.3.2. OVERLAY... Controlling the Hidden-Line Overlay of Element Borders On Data Plots

OVERLAY... Controlling the hidden-line overlay of element borders on data plots

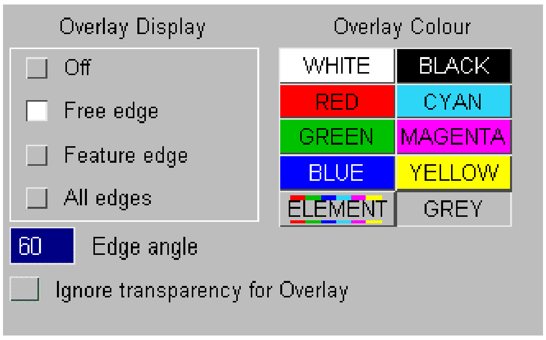
This figure shows the hidden-line overlay control panel.

This panel controls whether or not a hidden-line overlay is superimposed on plots, and its attributes (when drawn).

Overlay Display	Controls whether and how element overlay is drawn. It does not affect the current attributes.
Overlay Colour	Obviously sets the colour to be used. The default is white, but other standard colours can be chosen. **ELEMENT** colour

means use the **Overlay** colour of each element - see the **PROPS** box in [Properties: Controlling colour, drawing style, transparency, lighting attributes and overlay of entities](#).

Hidden-line overlays apply as follows:

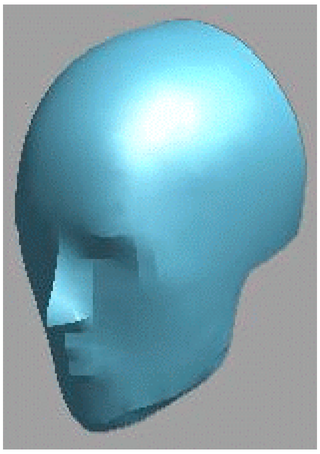
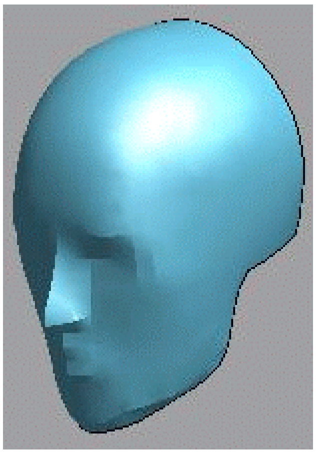
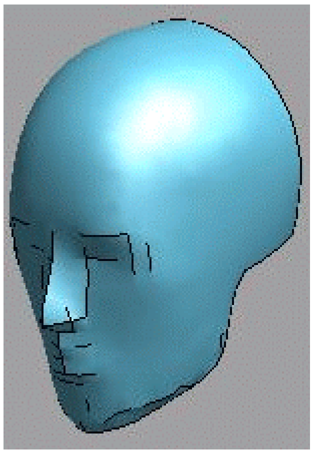


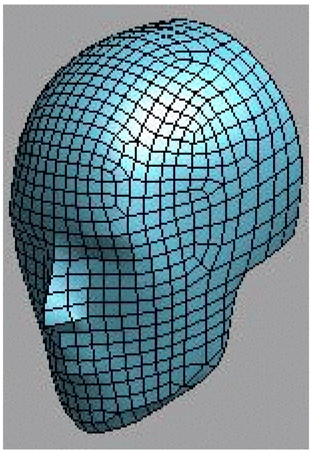
- Overlays affect all data plotting display modes, and also **SH** shaded (**GREYSCALE**) plots.
- Their colour is normally fixed, but using the **ELEMENT** option, in conjunction with the [**PROPS**](#) panel, permits any permutation of overlay colours to be used.
- The display of edges may be one of:

OFF	No element overlay is drawn
Free edge	The topological free edges are drawn
Feature edge	Free and "feature" edges are drawn
All edges	All element borders are drawn

The **Edge Angle** is the angle between adjacent facets at which an "edge" is deemed to occur. It affects both Feature edges and also smooth shading. The default of 60 degrees is satisfactory in most cases, but to obtain more edges reduce this value. Values approaching 180 degrees will eliminate edges altogether.

The effect of these various edge drawing options is shown in the four images below.

		
This image has no overlay at all	This image has only free edge overlay, which is drawn only at the element	This image has feature line overlay with an "edge angle" of 20 degrees,

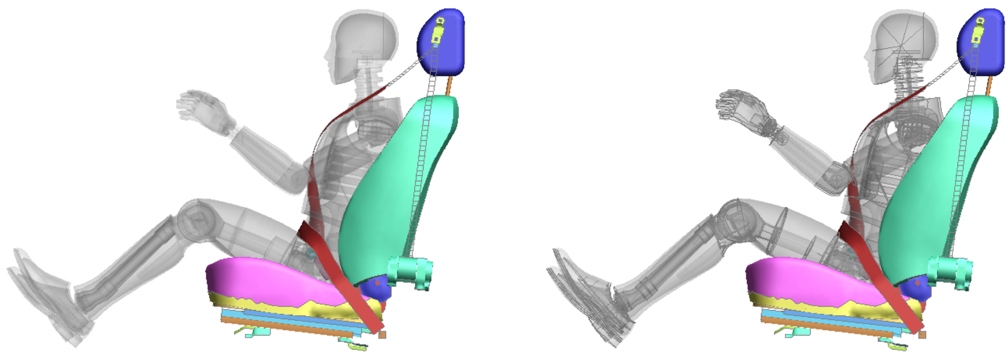
	edges at the back of the head.	showing detail around the mouth, nose and eyes.
		
This image shows all element borders.		

7.3.3.2.1. Overlay Transparency

Overlay Transparency

From D3PLOT 15.0 onwards, the overlay is by default drawn using transparency if the underlying entity has also been drawn using transparency.

This option can be used to turn off the transparency applied to the overlay so that D3PLOT reverts to it's pre-D3PLOT 15.0 behaviour.



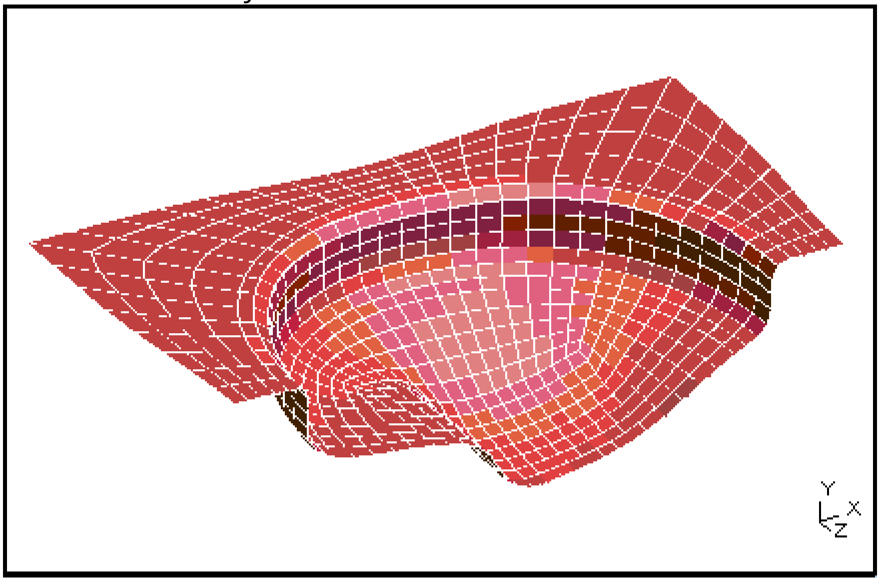
7.3.3.2.2. Controlling Overlay Quality In 3D Mode

Controlling overlay quality in 3D mode

Hidden-line removal in Z-buffered 3D graphics presents a problem to the programmer because the edges round elements will look "patchy" if they are drawn (correctly) co-planar with the element infill. Therefore overlay has to be raised slightly towards the observer (in the screen space Z dimension) to lift it above the surrounding elements, and the algorithm which calculates this "Z lift" dimension is usually satisfactory for all normal viewing parameters. However there are a few cases in which it can go wrong:

This image demonstrates the "patchy" overlay that can occur if the "Z lift" dimension is insufficient to raise it above the surrounding element infill. This does not normally happen, but it can occur if:

- Perspective is turned on
- And the perspective distance is very small



To fix this problem use < **`right ctrl`** > + < ***`mouse button`*** >. Moving the mouse up the screen will raise the overlay towards you, moving it down away. The mouse button used controls the speed of "Z lift" change:

< *Left mouse* >	Produces slow change
< *Middle mouse* >	Moves it more quickly
< *Right mouse* >	Makes large changes

If you subsequently revert to a more "normal" viewing distance you may need to reset the "Z lift" to get an acceptable image quality.

7.3.3.3. Properties: Controlling Colour, Drawing Style, Transparency, Lighting Attributes and Overlay of Entities

Properties: Controlling Colour, Drawing Style, Transparency, Lighting Attributes and Overlay of Entities

It is important to understand the distinction between "Properties" and "Settings" in D3PLOT:

Properties	Are attributes of a model, for example part colours.

Settings Are attributes of the programme and menu interface, for example data component.

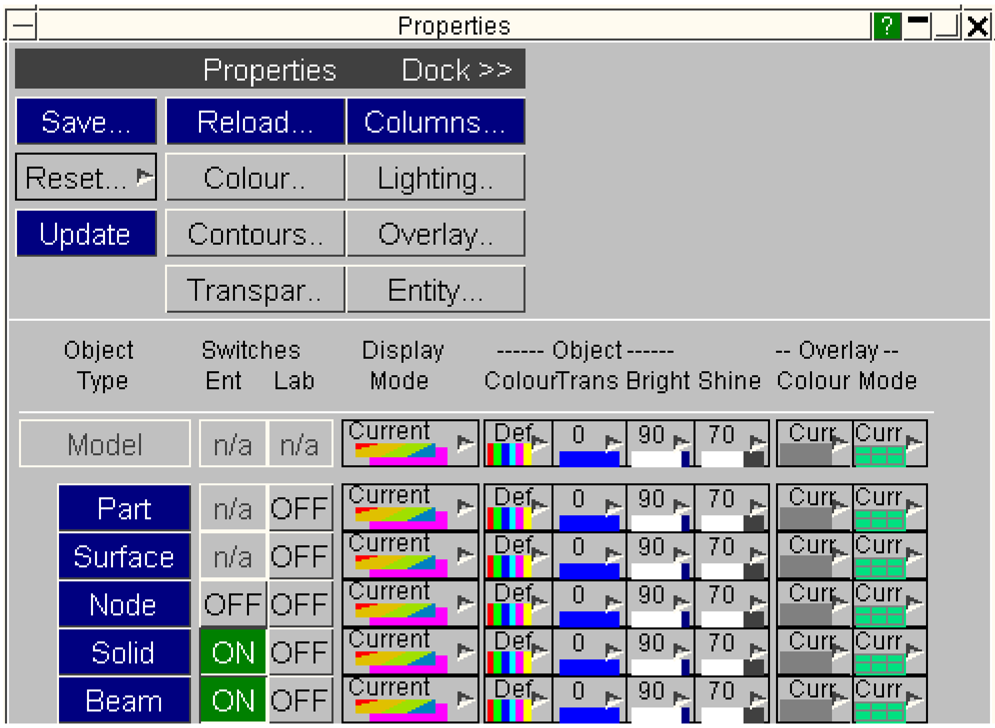
A fuller description of these differences is given in What is the difference between a Properties file and a Settings file? .

The **PROPS** box duplicates the colour setting capabilities of the **COLOUR** command above (which is kept for backwards compatibility), but provides many more capabilities for improving the visual properties of plots:

- Explicit visibility (blanking and entity switch) control.
- Mixed display modes (contoured/shaded/hidden/wireframe).
- The ability to label items selectively.
- Colour setting.
- Transparency setting.
- Lighting attributes (diffuse brightness and shininess).
- Overlay colour and style (solid/free edge/omitted).

These properties may also be changed using Quick-Pick.All of these capabilities are available at <model>, <category (eg part)> and <individual item> level; which makes it possible to tune plots for presentation to any degree. Model properties can be saved and restored, and even applied to different models: see OPTIONS... SAVE and RELOAD: Saving and Restoring Properties. Due to the width of the menu, the option is provided to **UNDOCK** the menu. This moves it from being docked in the menu area to floating in the Graphics area. **DOCK** will reverse this process.

From D3PLOT 11.0 onwards, property "states" (unrelated to data states) can be saved using the **Save P** button in the view panel - see Saved properties.



7.3.3.3.1. Selecting the Level at Which to Operate: <Model> <Category> <Entity>

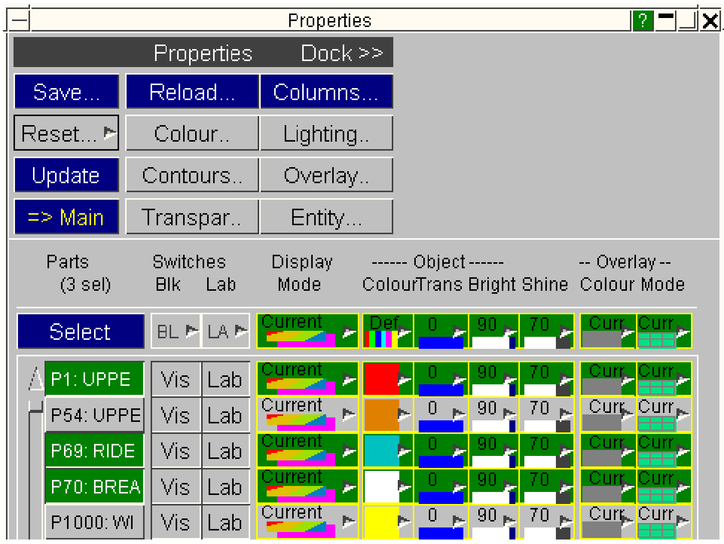
Selecting the level at which to operate: `<Model>` `<Category>` `<Entity>`

The **PROPS** box operates at two levels: the top one applies to the whole model, and all its constituent categories; the lower one applies to individual items in a category. In this example the user has chosen to operate on individual parts.

Selecting **PART** at the top level...

...results in the **PART s** sub-panel being displayed.

It is also possible to **SELECT** any permutations of items, using the standard D3PLOT selection process or by clicking on their id button. Selected items have their background changed to green (here parts 1, 3 and 4 have been selected).



Operations applied to the top row buttons are then applied to all selected items.

7.3.3.3.2. Columns.. Controls Which Attributes Columns are Shown

Columns.. Controls which attributes columns are shown

Because the **Properties** box can be configured to contain a lot of information, which can look confusing, the **Columns...** menu can be used to control which columns of attributes are shown. There are four of these, which may be turned on/off independently:

Entity/Label Switches

At Top level the **Entity** switch controls whether or not that category is drawn at all (see ENTITY Switching the Display of Entity Categories On/Off). The **Label** switch controls whether or not it is labeled at all (see ENTITY Switching the Display of Entity Categories On/Off). At sub-panel level the "Entity" column is replaced by **BLANK** : each element may be (un) blanked individually (see BLANK "Blanking" Controls the Visibility of Nodes and Elements). Similarly the **LABEL** switch may be applied individually to each element.

Display Modes

At top level each category, or the whole model, may be set via its popup menu to one of WIREFRAME, HIDDEN, SHADED (see Drawing Commands that Do Not Plot Data for what these mean) or Current (Elements are drawn in the currently selected display mode, whatever that may be). At sub-panel level each entity can be set separately in a similar way. It is possible to select any mode for any element, and to display them in combination. For example you could have a mixture of shaded, wireframe and contoured display in different areas of a mesh.

Solid-filled (ie **CT** or **SI**) contours are not shown on elements rendered in wire, hidden or shaded modes. However vector data (**LC** , **VEL** , **VEC** **,** **Criterion**) is superimposed on the current display mode of elements. For example velocity vectors may be drawn on shaded elements.

Object Attributes

As before changes may be applied to whole categories at the top level, and to individual or selected items at the sub-panel level. There are four "attributes":

Colour : Select an explicit colour, or return to default.

Transparency : Transparency only applies to 2D and 3D objects, and by default they are all totally opaque. However you may set any such entity's transparency on the range 0% (opaque) to 100% (fully see-through) in increments of 10%.

Brightness : Setting diffuse brightness

Shininess : Setting specular brightness

Lighting is discussed in more detail in The LIGHT panel below.

Overlay Attributes

Overlay Colour : Select an explicit overlay colour, or default.

Each entity maintains a separate overlay colour, distinct from its "current" colour. The **Default** overlay colour is the same as the native element colour, but the two are stored separately and may be quite changed independently.

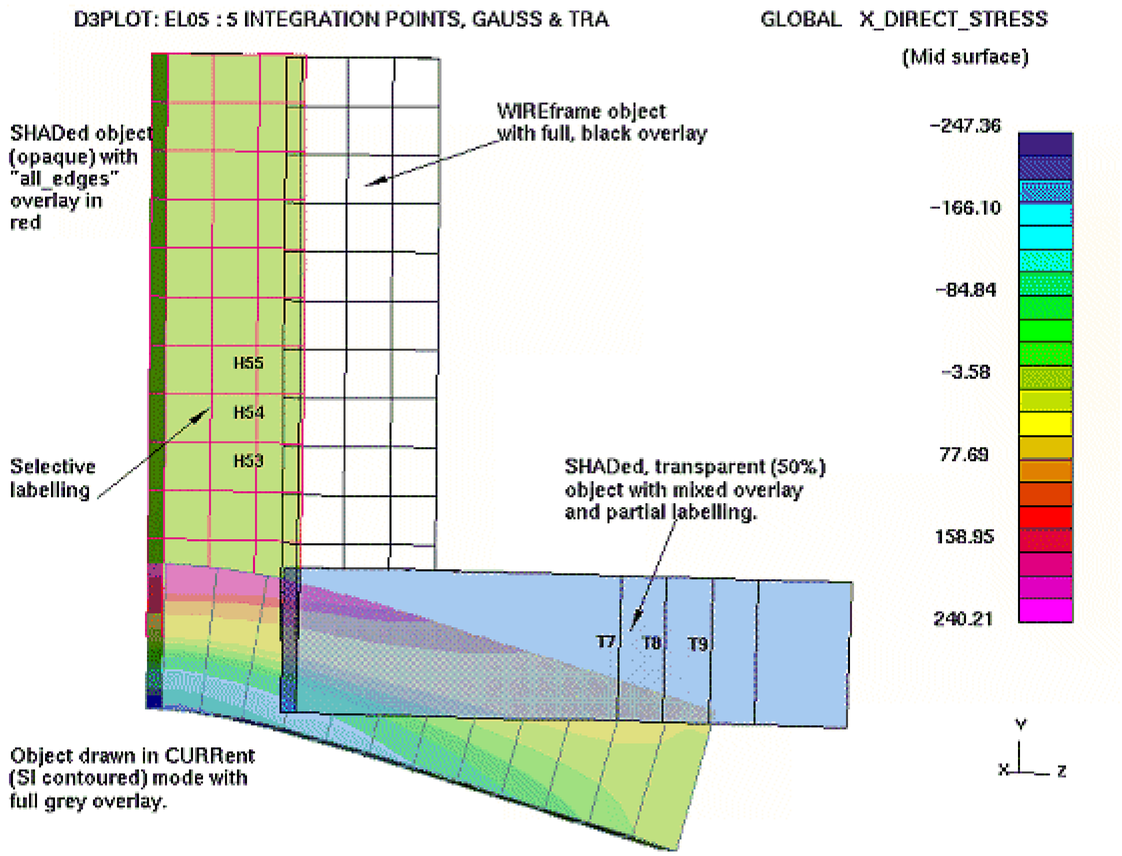
Overlay Mode :

Overlays may be drawn in one of three styles:

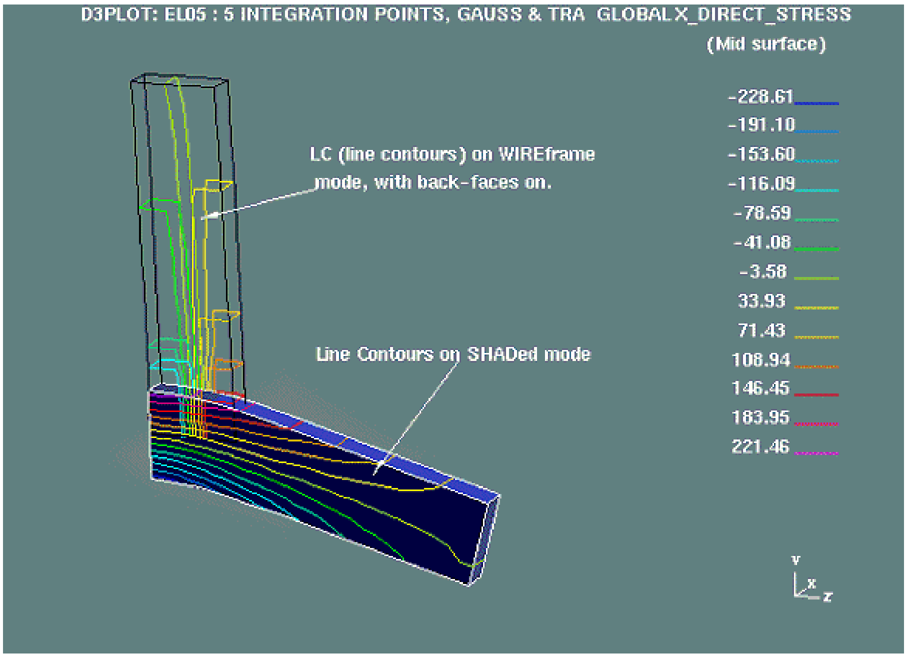
NO_OVERLAY: Not drawn at all.
FREE_EDGE: Only free edges are drawn.
ALL_EDGES: All edges are drawn.

When drawing the more restrictive of the main **DISPLAY_OPTIONS** , **FREE_EDGE** switch and the "local" element ones is applied: if either eliminates edges they will not be drawn. See OVERLAY... Controlling the Hidden-Line Overlay of Element Borders On Data Plots for more detail.

So what does all this mean? Here is an example which combines transparency, different modes of plotting, selective labeling and various overlays to show what can be achieved.



Here is another example which shows how "vector" mode plots (here **LC** , but also **VEL** , **VEC** & **Criterion**) can be superimposed on shaded and wire-frame rendering.



7.3.3.3.3. OPTIONS... SAVE and RELOAD: Saving and Restoring Properties

OPTIONS... SAVE and **RELOAD**: Saving and restoring properties

Setting up properties (colour, transparency, etc) for a model can be time-consuming, and it is useful to be able to preserve them for subsequent runs.

- **SAVE** writes the complete property list for the whole model into a "properties" (`.prp`) file.
- **RESTORE** can read this in at a later date and restore all the saved settings.

You can create any number of property files, each will save the current status, and these may be read in at any time to update the current display.

Property files can be applied to different (but similar) models:

- All information in a property file is written using ***external*** item labels.
- When read back in these labels are mapped onto the items in the current model that have these labels.
- If an external label is encountered that has no internal counterpart it is ignored.
- Similarly those items in the current model which don't appear in the properties file are left unchanged.

Therefore you can apply a properties file to any model and, provided that it is not too dissimilar to the model from which the file was written, the effect should be to restore your settings almost completely.

Reading and writing properties files elsewhere in D3PLOT

Properties files can also be written and read back in from the **UTILITIES, SETTINGS_FILE** panel. This is described in UTILITIES, SETTINGS_FILE .

They can also be read in at the same time as the model is input if the **Read PRP file** box is checked on the file input panel - see Open a Single Model

From release onwards they can also be read and written from the Saved properties option panel, see Export to file... saving properties to file

What is the difference between a Properties file and a Settings file?

A ***properties*** file contains information about the model properties: colour, transparency, labeling, visibility, display modes, etc. It is window independent.

A ***settings*** file contains information about the D3PLOT window settings: background colour, data component, contour levels, etc. It is model independent.

Properties and Settings files are forwards, but not backwards, compatible.

Both of these file types evolve with successive releases of D3PLOT. A newer format file will not read into an older version of the code, however an older format file may be read into a newer version of the programme and - usually - will function normally.

There are some minor exceptions: some overlay attributes in a properties file from release 9.0 of D3PLOT may not translate properly to a newer release. This is not so much because of file incompatibility, but rather because of changes to the inner workings of the code itself which make the "old" attributes invalid.

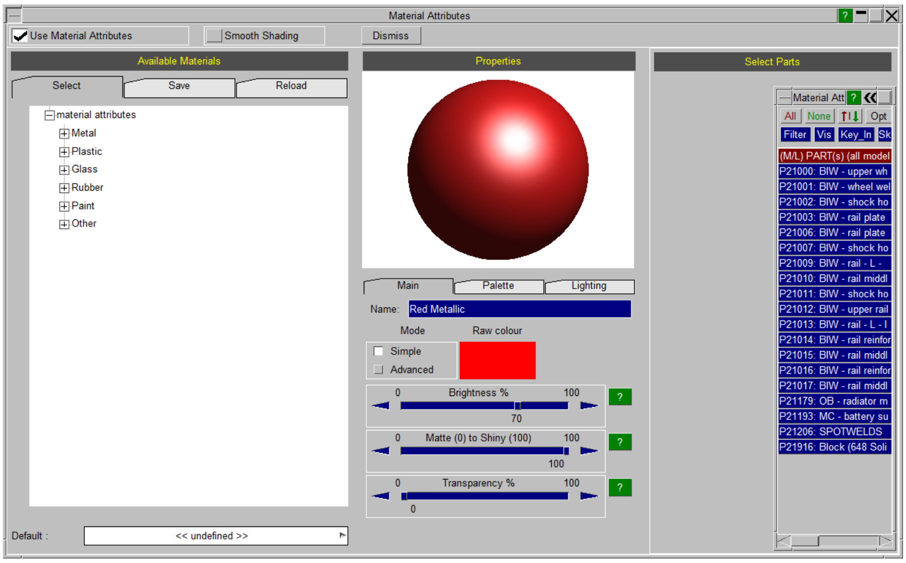
From D3PLOT 11.0 onwards the properties file is now both model and programme-independent, and in particular properties may be exchanged between D3PLOT and PRIMER. The format of the file and an explanation of its contents can be found in The Format of the Saved Properties (.prp) File

If you have problems with incompatibility please contact Oasys Ltd for help and advice.

7.3.3.4. Material Attributes

Material Attributes:

The Material Attributes panel offers increased control over part-specific colour and lighting. Materials with specific lighting properties can be saved and applied to parts to tailor the model's appearance.



7.3.3.4.1. Available Materials

Available Materials

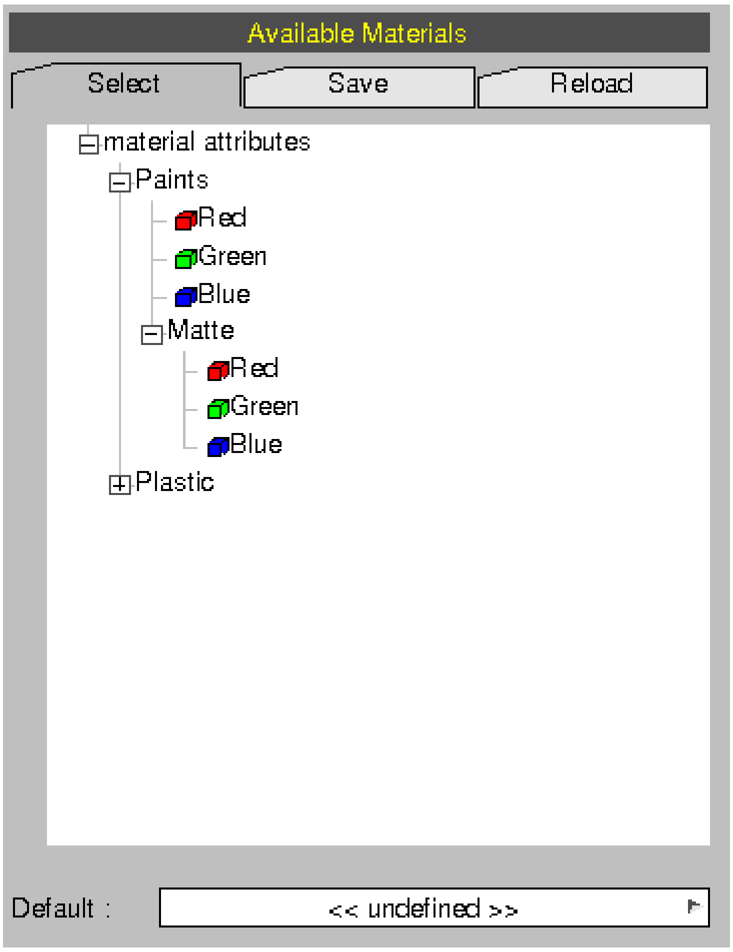
The *Available Materials* area on the left-hand side of the panel lists user-defined *Material Attributes* arranged by *Category* .

Categories act as a means to organise *Material Attributes* . In the image to the right, "Paints" is a category and "Red" is a material attribute belonging to "Paints". Categories can also contain sub-categories to further organise material attributes. In the same image "Matte" is a sub-category of "Paints"

Each material attribute has a corresponding set of properties which control its colour and lighting. Once these properties have been set, the material attribute can be applied to any part in the model. The link between parts and material attributes is retained for the duration of a D3PLOT session, so if the material attribute is modified the changes are reflected in the appearance of the parts to which that attribute has been applied. The link does not persist beyond the current session of D3PLOT.

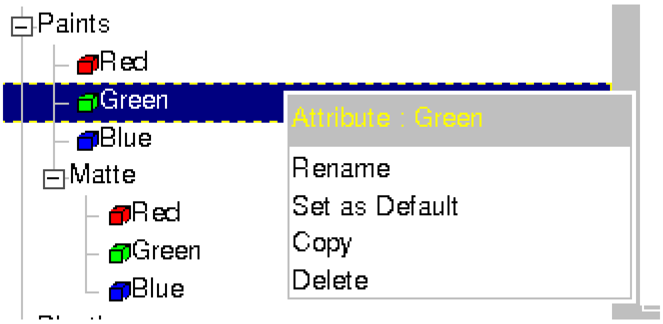
Material attributes and categories can be saved for future use using the "Save" tab. This will output an XML file listing all of the material attributes and their corresponding properties. However, this XML file is independent of the model and consequently will not record the link between parts and material attributes.

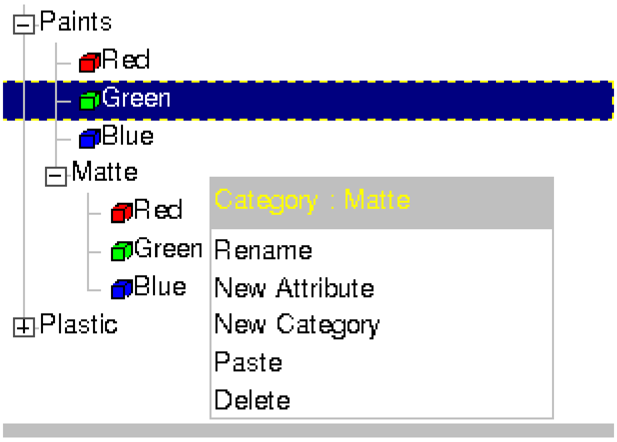
A previously saved XML file can be read back into the panel by using the "Reload" tab.



Any material attribute can be copied and pasted allowing variants, e.g. different colours of paint, to be quickly created.

Finally, a material attribute can be set as the "Default" by right-clicking on it and selecting "Set as Default". This will apply the selected material attribute to all parts within the model that do not currently have a material attribute applied to them.





7.3.3.4.2. Material Properties

Material Properties

Material attributes are modified using the central *Properties* part of the panel. This area of the panel includes a sphere showing a preview of the material attribute appearance.

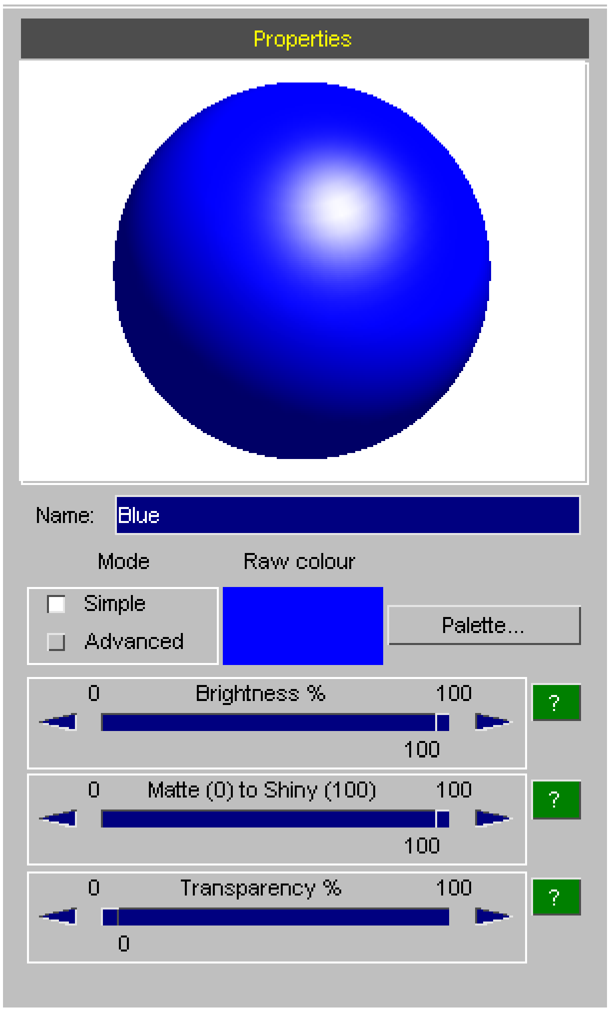
The material attribute name can be edited in the "Name" textbox.

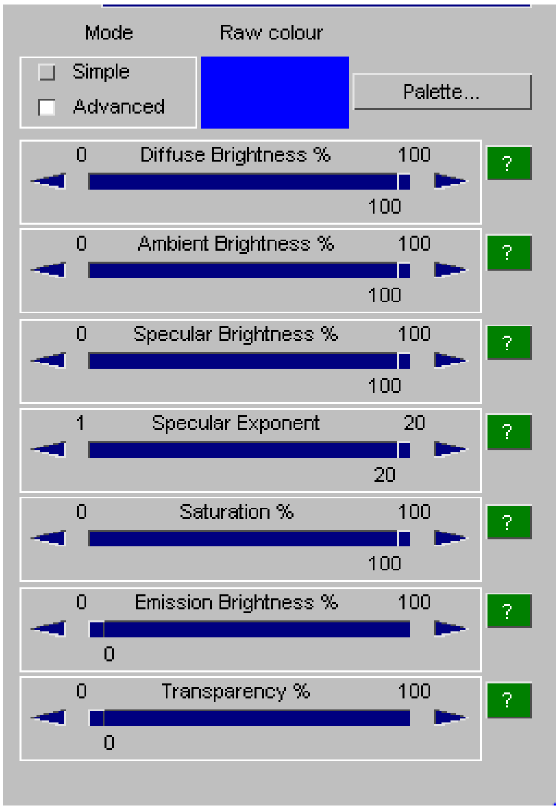
The material attribute colour can be modified by pressing the "Palette" button which brings up sliders to control RGB values or hue angle, saturation and light.

Finally, the properties can be adjusted using a series of sliders. "Simple" mode provides a reduced number of sliders simplifying the modification of properties, but reducing the subtlety of adjustments that can be made. The "Advanced" mode exposes all the controls.

The table below documents how the properties between "Simple" and "Advanced are linked to one another

Simple	Advanced
Brightness %	Diffuse Brightness %
	Ambient Brightness %
Matte (0) to Shiny (100)	Specular Brightness %
	Specular Exponent
	Saturation %
N/A	Emission Brightness
Transparency %	Transparency %



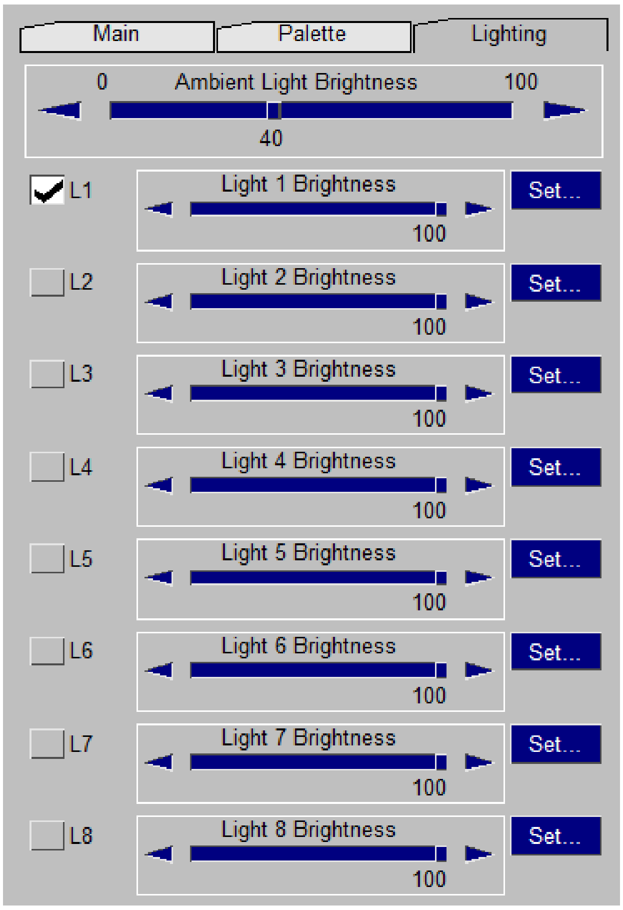


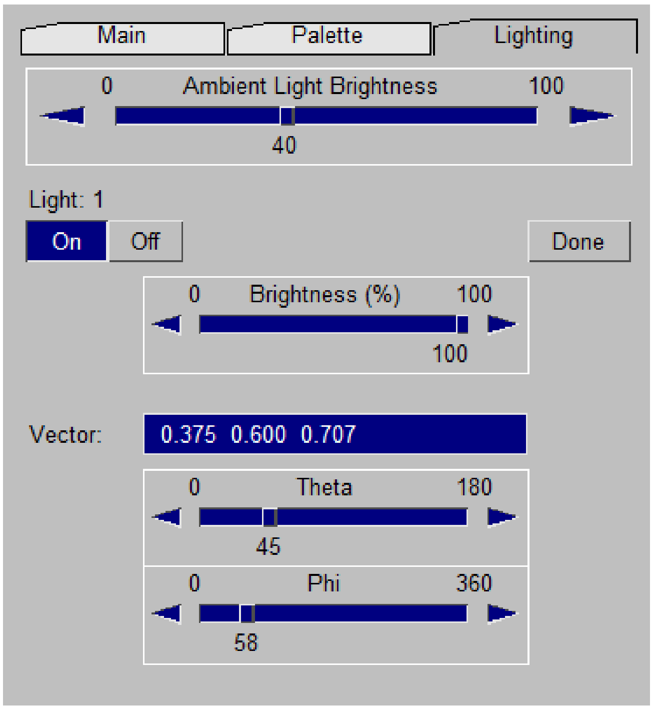
7.3.3.4.3. Lighting

Lighting

In D3PLOT 17.0, we've added a Lighting panel to the Properties panel. Within this panel you will be able to control all 8 available lights brightnesses and positions instead of having to switch over to the old lighting panel.

Each light has a main switch, brightness slider and a "Set..." button which when pressed takes you to that particular light's sub-panel. Within this sub-panel you can still control the brightness of that light but also it's vector. This can be controlled either by imputting data into the text box however it's been built so that you can now more interactively control the position via the "Theta" and "Phi" sliders (spherical coordinates). This means that you no longer have to work out a sensible light vector or position, you can just experiment with the slider controls to get a light in a position that looks about right for your model. These sliders have also been placed within the original Lighting panel.



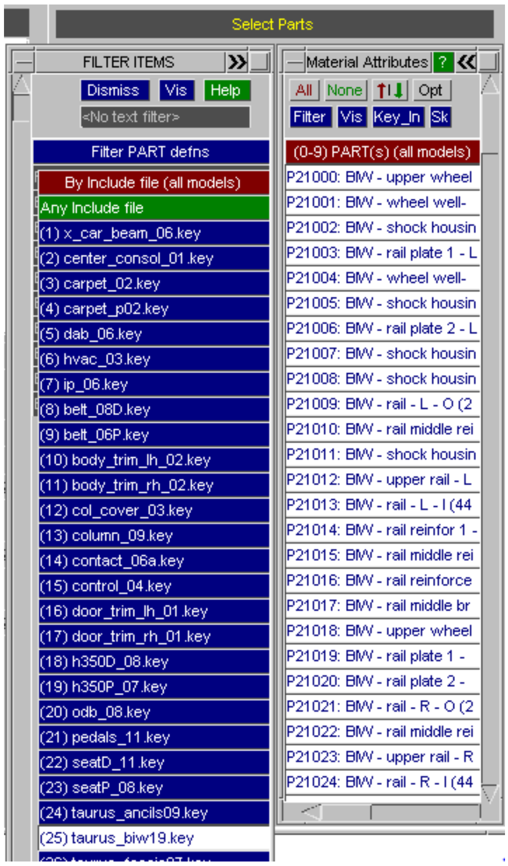


7.3.3.4.4. Selecting Parts

Selecting Parts

Using the entity panel on the right-hand side of the panel lets you assign attributes to particular parts. This is done simply by selecting the desired material attribute in the left-hand part of the panel then selecting the parts to which that attribute should be applied in the object menu. Once the parts are highlighted in the object menu the material attribute will be applied, it is not necessary to "apply" the selection. The model view will update in real time.

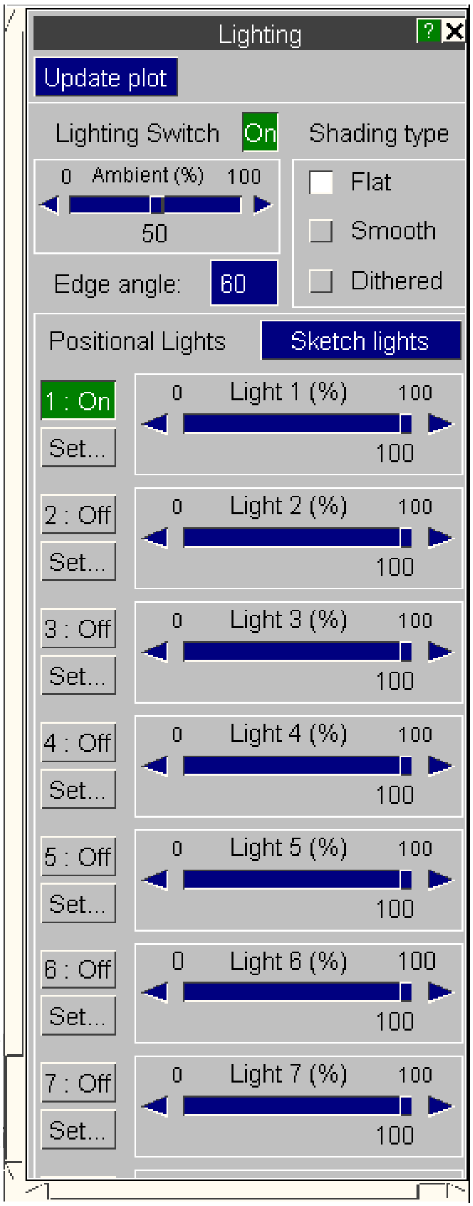
The parts list can be filtered down to the parts that you want to select using the standard object menu functionality. Filtering by Include file can be a particularly effective approach, if your model uses them. For example, in the case of a car model, this allows you to quickly select all of the parts which belong to the "Body in White" and assign a Material Attribute to only these parts (shown below in blue)


Select Parts
FILTER ITEMS
Dismiss
Vis
Help
<No text filter>
Filter PART defns
By Include file (all models)
Any Include file
(1) x_car_beam_06.key
(2) center_consol_01.key
(3) carpet_02.key
(4) carpet_p02.key
(5) dab_06.key
(6) hvac_03.key
(7) ip_06.key
(8) belt_08D.key
(9) belt_06P.key
(10) body_trim_lh_02.key
(11) body_trim_rh_02.key
(12) col_cover_03.key
(13) column_09.key
(14) contact_06a.key
(15) control_04.key
(16) door_trim_lh_01.key
(17) door_trim_rh_01.key
(18) h350D_08.key
(19) h350P_07.key
(20) odb_08.key
(21) pedals_11.key
(22) seatD_11.key
(23) seatP_08.key
(24) taurus_ancils09.key
(25) taurus_biw19.key
Material Attributes
?
All
None
Opt
Filter
Vis
Key_In
Sk
(0-9) PART(s) (all models)
P21000: BIW - upper wheel
P21001: BIW - wheel well-
P21002: BIW - shock housin
P21003: BIW - rail plate 1 - L
P21004: BIW - wheel well-
P21005: BIW - shock housin
P21006: BIW - rail plate 2 - L
P21007: BIW - shock housin
P21008: BIW - shock housin
P21009: BIW - rail - L - O (2
P21010: BIW - rail middle rei
P21011: BIW - shock housin
P21012: BIW - upper rail - L
P21013: BIW - rail - L - I (44
P21014: BIW - rail reinfor 1 -
P21015: BIW - rail middle rei
P21016: BIW - rail reinforce
P21017: BIW - rail middle br
P21018: BIW - upper wheel
P21019: BIW - rail plate 1 -
P21020: BIW - rail plate 2 -
P21021: BIW - rail - R - O (2
P21022: BIW - rail middle rei
P21023: BIW - upper rail - R
P21024: BIW - rail - R - I (44

7.3.4. The LIGHT Panel

The LIGHT panel

The lighting model in D3PLOT is a simplified version of the diffuse and specular models found in most graphics hardware [(1)], and in standard computer graphics texts [(2)]. It is a compromise between speed and appearance that is adequate for rendering engineering plots and simple presentational material, but its results fall short of those from a ray-tracing package: shadows and reflections are not provided, and surface properties are crude.



Geometry can be exported to external rendering packages via the **UTILITIES** , **VISUALISATION** menu for those cases where higher quality images are required.

The lighting model is best thought of as having three parts:

1. The light sources .
2. The way in which the objects are shaded .
3. The material properties of the lit objects .

The Light panel ontrols the following lighting and shading attributes:

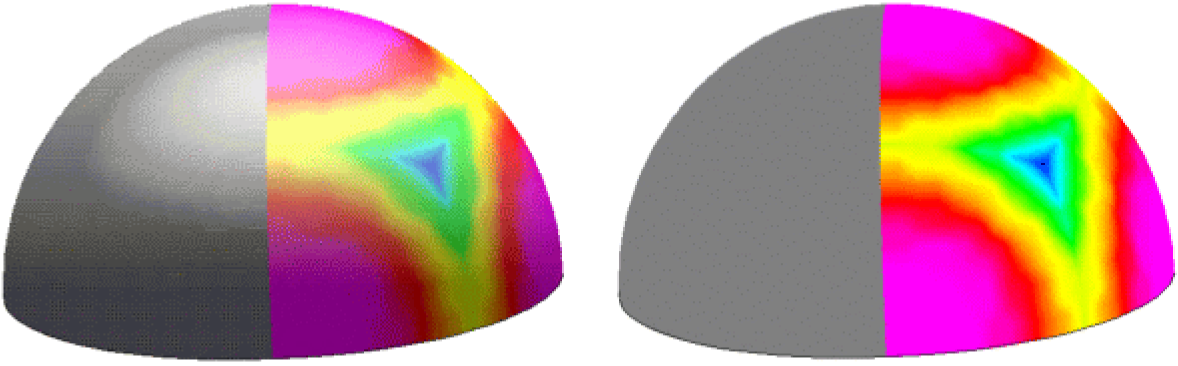
Lighting switch :	Global on/off for lig
Point light sources :	On/off switch, posit (model/screen).
Ambient light :	Intensity.
Shading type :	Flat / Smooth / Dithered.
Edge angle:	Angular difference limit for smooth shading across adjacent facets.

7.3.4.1. Lighting Switch Global Lighting On/Off Switch

Lighting Switch Global lighting on/off switch

By default lighting is ***ON*** , and a plot implying lighting will use the current attributes. If lighting is turned ***OFF*** all formerly lit elements will be drawn at the full colour intensity with no shading effects.

Turning the switch on/off does not change stored lighting or shading attributes in any way.



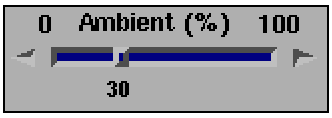
Lighting Switch ON
Lighting Switch OFF

These two examples show how a hemisphere, half **SI** contoured and half **SH** shaded, responds to the lighting switch. Note how turning lighting off destroys any perception of shape or depth.

(The effect on the right could also be achieved with lighting ***ON*** by setting the **Ambient** light to 100% and having no point light sources active.)

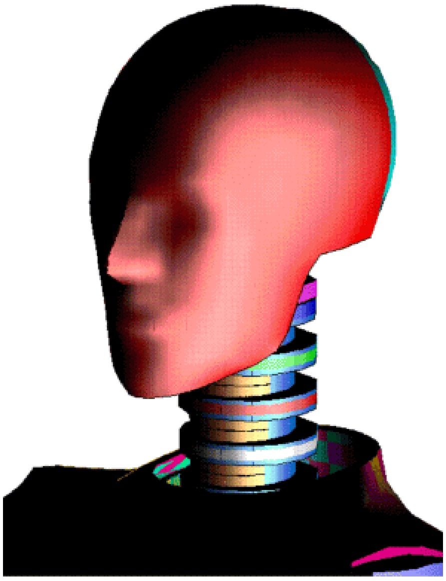
7.3.4.2. Ambient Light The %age Of "Black Body" Light

Ambient light The %age of "black body" light



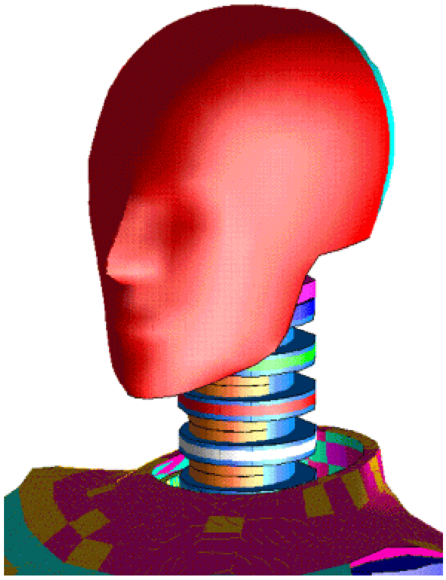
If only point light sources are used this simple lighting model produces images reminiscient of pictures from space: strongly directional lighting with no detail in shadow areas. Therefore the concept of **Ambient** light, analagous to "black-body" radiation coming from the universe at large, is added to fill on lowlight areas.

This is not realistic in itself, but in practice most scenes have their shadow and low-light areas filled in by reflected light from walls, floors, etc; and it provides a good approximation to this.



This image has a single directional light to the right, and has the **Ambient** light set to 0%.

Note how the lowlight areas are extremely dark and contain no detail.



Here is the same image with the **Ambient** light set to 40%.

This has filled in the lowlight areas to some degree, but a higher value still is needed to illuminate some areas.

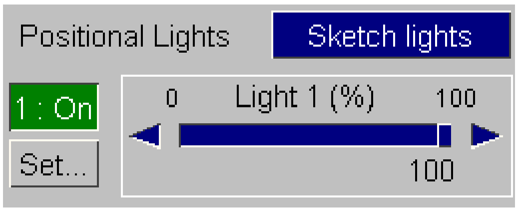
Why bother with ambient light? Why not just use more (and more realistic) point light sources?

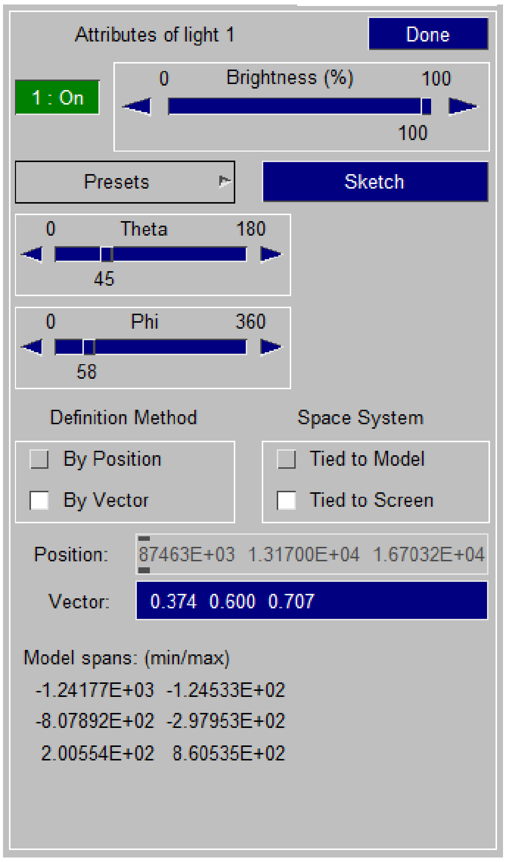
The short answer is speed and simplicity. Ambient light is cheap to compute and easy to define, whereas adding point lights slows down image redraw speed. And, as any photographer will testify, getting the position and intensity of multiple light sources correct is not as simple as it seems. However you have 8 light sources to experiment with: feel free! (But note that some graphics hardware may not operate correctly, or may run slowly, with more than two light sources.)

The default in D3PLOT is a single directional light above and to the right of the observer's position, and an Ambient light level of 40%.

Point light sources (positional & directional)

Up to 8 point light sources may be defined, each of which is independently controllable. At the top level of the lighting panel they may be turned on/off and have their intensity set from 0 - 100%. To change any other attributes use the **SET...** button.





SET... Setting detailed light attributes

The following detailed attributes for each light may be set:

OFF/ON Turns this light on or off.

Brightness Sets the light brightness in the range 0 - 100%. (nb: it is more efficient to turn a light **OFF** than to use 0% **Brightness** .)

PRESETS >

Provides some pre-computed positions for lights.

These may not give exactly the locations you want, but they can form a good starting point.

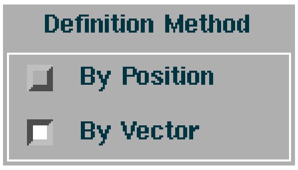
(The default light in D3PLOT, light 1, is positioned at "Right Shoulder".)

Theta/Phi

Spherical Coordinates control for the Vector of the specific light you are currently on.

This lets you experiment with the direction in which that light is facing more easily than selecting a specific vector via the textbox.

Definition Method Positional or Directional (by vector)



Example of Positional light

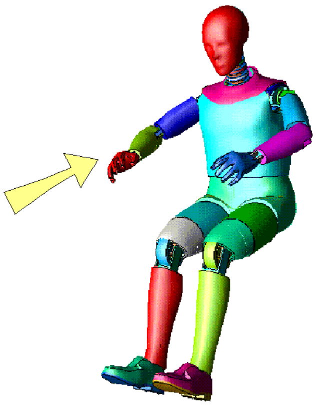
A "positional" light is defined by its location in space, and this method is generally used for lights close to the object.

This example shows a dummy with a positional light in its lap. (The light itself has been added artificially here, it would not normally be visible on a plot.)

Because the light is "local" it (correctly) does not illuminate the tops of the arms or the front of the legs.

Positional lighting is slightly more expensive than directional to compute, but is necessary if the true effects of lights close to the object are to be simulated.

Example of Directional light



A "directional" light is defined only by its vector, and assumed to be infinitely distant (like the sun).

This is the same model as above, but now the light is directional, defined by a vector pointing through the previous light source position towards the dummy. (The vector arrow has been added artificially, it would not appear on a plot.)

The differences are clear: the whole of the front of the dummy has been illuminated, as have the surfaces which previously were dark.

This is a little bit cheaper to compute than the positional equivalent, as "local" vectors from facet to light source need not be computed.

Defined lights as Tied to Model and Screen space systems

Light Tied to Model space

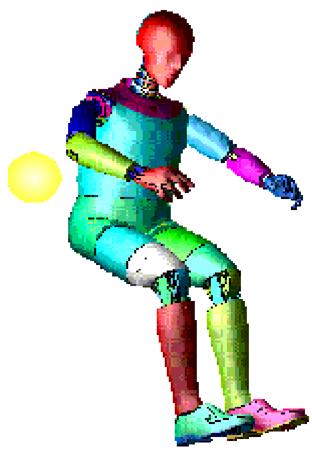
<== **Before** ..transform .. **After** ==>

When a light is tied to **Model** space it is transformed along with the object.

Light Tied to Screen space

<== **Before** ..transform .. **After** ==>

When a light is tied to **Screen** space it remains fixed in space as the object is transformed.

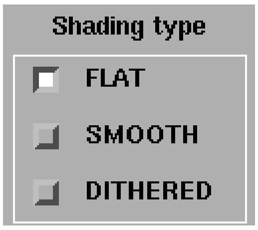


(In these examples the light sources have been added artificially to illustrate their positions. They would not appear on an actual plot.)

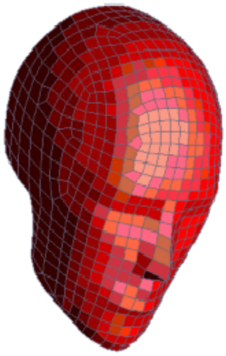
7.3.4.4. Shading Type:

Shading type:

Flat, **Smooth** and **D ithered** . This controls how facets are shaded, which in turn affects the appearance of curved surfaces.



FLAT shading



The outward normal of each element face is calculated and used to determine a single lighting value. This is applied to the whole face giving the faceted appearance shown here.

This is quick to compute and, with a fine enough mesh, gives acceptable image quality.

(The mesh overlay has been added here to emphasise that each facet has a single flat shade.)

SMOOTH shading

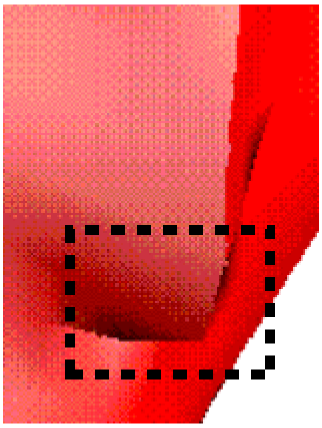
The outward normals at each vertex are averaged, making it possible to vary lighting smoothly across a surface. This technique is known as "Gouraud shading", and is only available in 3D graphics mode.

DITHERED shading (2D graphics mode only)

Because gouraud shading is not provided by 2D graphics drivers (or laser plotters) the technique of "dithering" has to be used to produce genuinely smooth shading under 2D graphics. This is done by drawing adjacent pixels in different colours to achieve an intermediate shade - trading of spatial against colour resolution.

Dithered shading is ignored in 3D mode, and should only be used in 2D mode if you are prepared to accept the longer computation and display times involved.

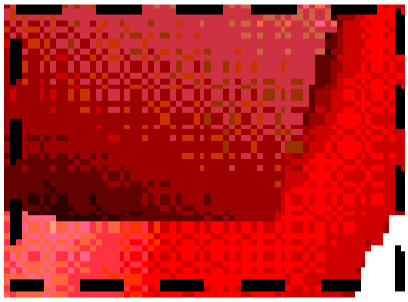
What dithering actually does, and how to fix problems that may arise from its use: (2D mode only)



On the left is an enlarged area of the dithered equivalent of the image above showing the bottom tip of the nose.

The dither pattern is just visible in the image on the left, and if magnified further (right) it becomes obvious.

There are only five different colours used in this image, yet it has been possible to show a wide range of shades. However spatial resolution has been lost in favour of colour range.



Dithering can occasionally cause problems when images are captured from the screen or laser-plotted. In particular there can occasionally be a "heterodyning" (beating) between the spatial resolution used for dithering on the screen and that used by the subsequent display device. This can show up as an apparent chessboard of large lighter and darker squares on the image, or as light/dark bands. Some software packages for manipulating bitmaps may show similar effects.

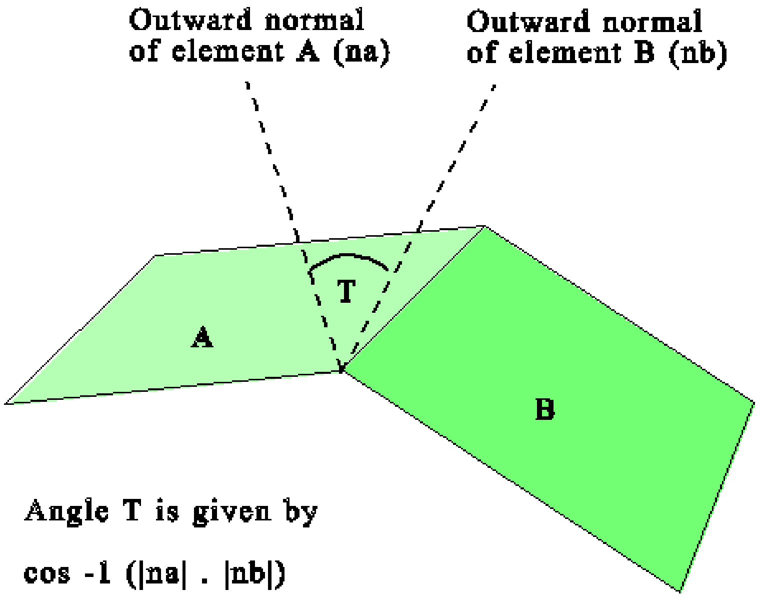
If this happens you may be able to fix it with one of the following:

- Try generating the image with a different screen window size, and hence a different scale. This may be enough to stop the "beating" effect.
- If you have been working at 8 bit-plane resolution, but your screen supports 24 bit-plane "true colour", then use that instead. At 24 bit-plane depth all possible colours that the human eye can resolve (2^{24} = 16,777,216) are available, and dithering is not required. The image will look better too.

7.3.4.5. Edge Angle:

Edge angle:

The angle at which smooth edges become sharp.



When smooth shading it makes sense to preserve some "sharp" angles, since most real-life objects have sharp as well as smooth edges.

This is done by computing the angle between the outward normals of adjacent facets at vertices, and only averaging if this is less than the current **Edge angle** . This effect is evident in the images of the head above: the ridge of the nose is "sharp" whereas the rest of the face is "smooth".

Edge angle can lie in the range 0 - 180 deg, (values > 90 deg are significant for 3D elements). The default of 60 degrees looks reasonably natural for most objects. 0 degrees is equivalent to Flat shading, and 180 degrees will eliminate all sharp edges.

7.3.4.6. The Light Sources Themselves

The light sources themselves

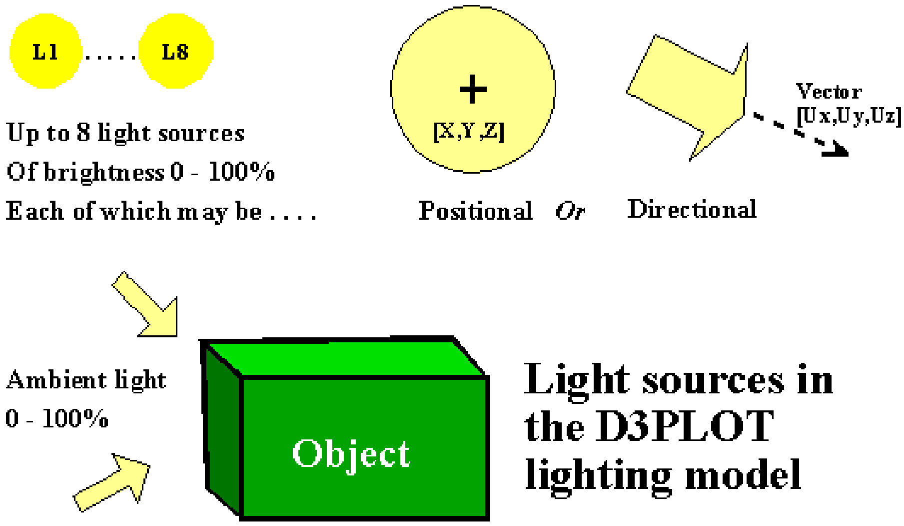
Up to 8 point light sources may be defined, each of which is independently controllable. At the top level of the lighting panel they may be turned on/off and have their intensity set from 0 - 100%. The OpenGL standard requires all implementations to handle 8 light sources, although it is often the case that only one or two cab be handled in hardware: don't be surprised if setting up many lights causes a sudden and dramatic reduction in graphics speed!

To change any other attributes use the **SET...** button. There are various preset options, and you can also specify your own light source positions or vectors.

Each may be "positional" (located at a specific point in space), or "directional" (infinitely distant along a vector). "Positional" light sources are more expensive for the hardware to calculate, and will slow down drawing, so only use them if you need to simulate the effects of a light close to your model.

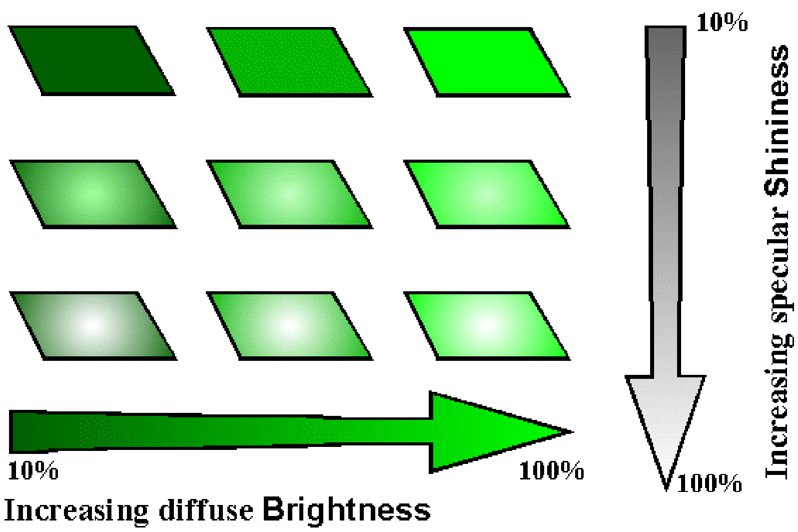
Light sources can be fixed in screen space (stay fixed when the model moves), or attached to model space (move with the model). There is no significant difference in graphics speed between the two options.

All lights are white.



The "material" properties of the lit objects

Brightness and **Shininess**



Brightness Controls how light or dark the colour of the object is when illuminated, but the effect is to add matt colour (not whiteness, which would make it look shiny).

Shininess Adds white highlights, but no colour, to make the object look shiny.

These attributes (of the object, not the lights) are set in the **PROPS** box - see Properties: Controlling colour, drawing style, transparency, lighting attributes and overlay of entities .

1. OpenGL Programming Guide. Neider, Davis, Woo (ISBN 9 780201 632743)

2. Computer Graphics Principles and Practice. Foley, van Dam, Feiner, Hughes (ISBN 0 201 12110 7)

7.4. DATA COMPONENTS - BASIC

DATA COMPONENTS - BASIC

All of the data plotting commands described in Displaying geometry and results use the **DATA COMPONENT** menu.

Data components and all of the settings associated with them are stored on a per window basis. If D3PLOT has multiple graphics windows open then any changes/selections that are made are only applied to the windows which have their `W1 .. Wn` tabs set.

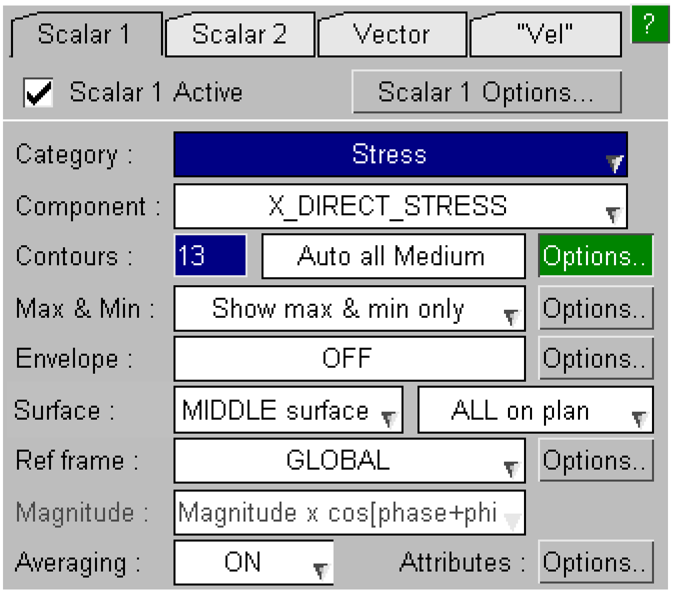
The contents of the data component menu always reflect the settings for the 1st window tab that is currently selected.

By default D3PLOT will plot data for a single data component "SCALAR 1", for more information on plotting multiple data components see DATA COMPONENTS - ADVANCED.

7.4.1. Selecting Data Components

Selecting Data Components

D3PLOT can access and display nearly 400 different data components. The data components available will depend on the model and the options selected in the LS-DYNA input file. For some data components D3PLOT will also need to read data from additional files.

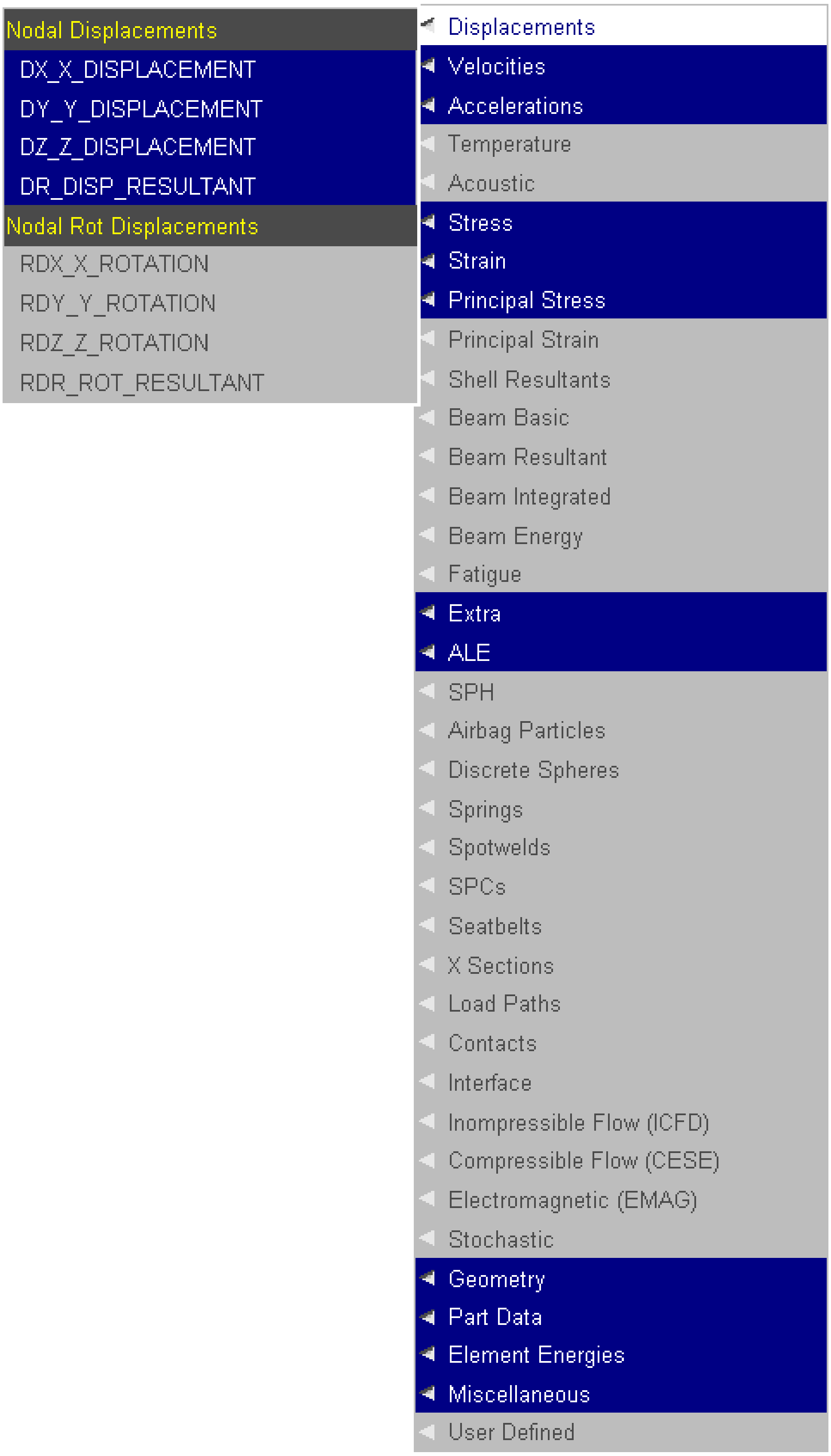


7.4.1.1. Category

Category

All of the components are grouped into a number of generic categories.

The number of contour levels, colours and ranges are (see LEVELS... Setting number of contour levels, their ranges, colours and number format) stored separately for each component category.



The following table gives a brief description of each data category and the LS-DYNA options needed to output the data for that category to be available in D3PLOT. Unless indicated all of the results are read from the D3PLOT (*.ptf) file.

Category	
Displacements Velocities Accelerations	Translational data components automatically generated for all nodes. Rotational components available for NASTRAN results read from OP2 file.
Temperature	Temperature and flux data components, controlled by THERM on *DATABASE_EXTENT_BINARY
Acoustic	Read from D3ACS file generated by *FREQUENCY_DOMAIN_ACOUSTIC_FEM
Stress	Symmetrical Stress tensor results, controlled by SIGFLG on *DATABASE_EXTENT_BINARY
Strain	Symmetrical Strain tensor results, controlled by STRFLG on *DATABASE_EXTENT_BINARY
Principal Stresses	Stress tensor results, controlled by SIGFLG on *DATABASE_EXTENT_BINARY
Principal Strains	Strain tensor results, controlled by STRFLG on *DATABASE_EXTENT_BINARY
Shell Resultants	Shell resultant forces and moments, controlled by RLTFLG on *DATABASE_EXTENT_BINARY
Beam Basic	Beam forces and moments, generated for all beam elements
Beam Resultant	Additional moment and energy components for resultant beams (Belytschko-Schwer), controlled by BEAMIP on *DATABASE_EXTENT_BINARY
Beam Integrated	Additional axial and shear stress/strain components for integrated beams (Hughes-Liu), controlled by BEAMIP on *DATABASE_EXTENT_BINARY
Beam Energy	Beam energies generated by NASTRAN and read from OP2 file.
Fatigue	Read from D3FTG file generated by *FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE
Extra	Additional time history variables for solids and shells, controlled by NEIPH and NEIPS on *DATABASE_EXTENT_BINARY
ALE	
SPH	Additional SPH data components generated automatically for *ELEMENT_SPH definitions. In addition to these components SPH elements also output stress and strain tensor results which can be contoured using the STRESS and STRAIN component categories
Airbag Particles	Airbag particle data components generated automatically for *AIRBAG_PARTICLE definitions.

Discrete Spheres	Additional Discrete Sphere data components generated automatically for *ELEMENT_DISCRETE_SPHERE definitions. In addition to these components Discrete Sphere elements also output stress tensor results which can be contoured using the STRESS component categories
Springs	Spring forces read from LSDA (binout) file, controlled by *DATABASE_DEFORC
Spotwelds	Spotweld forces read from LSDA (binout) file, controlled by *DATABASE_SWFORC and *DATABASE_DCFAIL
SPC's	SPC reaction forces read from LSDA (binout) file, controlled by *DATABASE_SPCFORC
Seatbelts	Seatbelt, slipring and retractor results read from LSDA (binout) file, controlled by *DATABASE_SBTOUT
X-Sections	Cut section forces and moments read from LSDA (binout) file, controlled by *DATABASE_SECFORC
Load Paths	Load path forces, uses cut section forces and moments read from LSDA (binout) file, controlled by *DATABASE_SECFORC
Contacts	Contact segment forces, read from CTFILE. Output controlled by SPR and MPR on *CONTACT, *DATABASE_BINARY_INTFOR and "S=" command line option.
Interface	Other interface force results read from BLSTFOR, FSIFOR, CPMFOR and DEMFOR files.
Incompressible Flow (ICFD)	Generated by the new Incompressible flow solver in LS-DYNA
Compressible Flow (CESE)	Generated by the new Compressible flow solver in LS-DYNA
Electromagnetic (CESE)	Generated by the new Electromagnetic flow solver in LS-DYNA
Stochastic	Generated by the new Stochastic particle solver in LS-DYNA
Geometry	Volume, relative volume and thickness (controlled by ENGFLG on *DATABASE_EXTENT_BINARY)
Part Data	Part bases energies, velocities, momentum and material properties. For the material properties to be available D3PLOT also needs the ZTFILE (*.ztf) file generated by PRIMER.
Element Energies	Element energies generated by NASTRAN and read from OP2 file.
Miscellaneous	
User Defined	User defined data components (see User Data).
Metal Forming	Components relevant to metal forming (see METAL FORMING).

Material Properties	Similar to but a more refined version of the 'Extra' category. Permits selective contouring of parts for which specific history variables are applicable

For these categories to be available D3PLOT also needs the ZTFILE (*.ztf) file generated by PRIMER as the output files generated by LS-DYNA do not contain enough information to draw these entities. A " **d3plot.components** " file (include in the installation package) is also required.

As the process of opening and scanning the contents of large LSDA files can be very slow on network disks D3PLOT uses a separate thread for opening the LSDA file. While the file is still being opened and scanned these components will remain greyed out but the rest of the menus in D3PLOT will still be available. When the thread has finished opening the file the components will become available

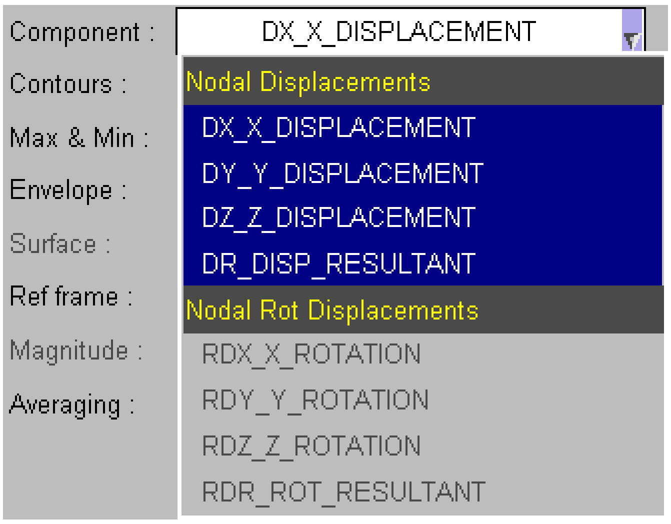
For these categories to be available a " **multiphysics.components** " file (included in the installation package) is also required.

For these categories to be available the " **mat_prop.csv** " file (included in the installation package) and the ZTFILE (*.ztf) file generated by PRIMER are both required. To use a customised version of the file mat_prop.csv, the preference `d3plot*mat_prop_location` can be used to specify a different file path.

7.4.1.2. Component

Component

After selecting a data category and component the Component button can be used to quickly swap between other components in the same category.



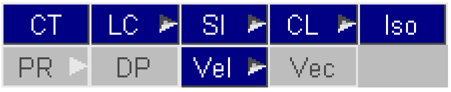
7.4.1.3. Plotting Modes

Plotting Modes

Whenever the data component changes the plotting modes available will update to those that are applicable for that component.

For more details on the different plotting options see [Drawing commands that plot data](#)

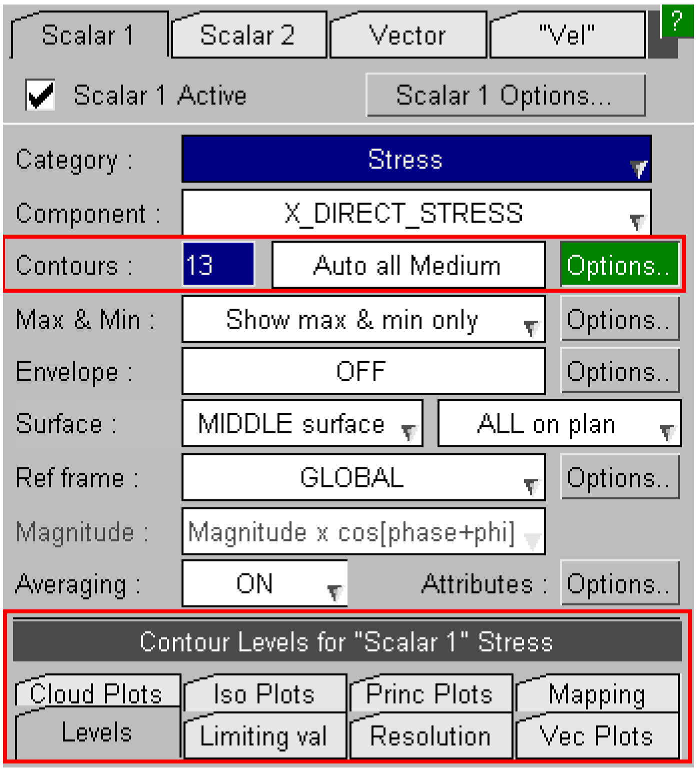
The options shown at the bottom of the main window show the plotting options available for the data component in the 1st currently selected window.



The options at the top of each graphics window show the plotting modes available for the data component currently selected in that window.

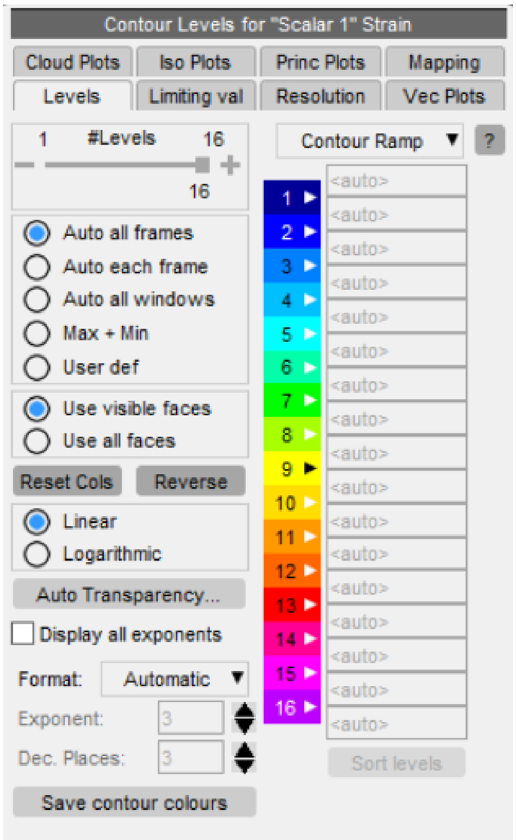
7.4.2. Contour Options

Contour Options



7.4.2.1. LEVELS... Setting Number of Contour Levels, their Ranges, Colours and Number Format

LEVELS... Setting number of contour levels, their ranges, colours and number format



By default contouring is set to have:

- 13 levels
- Automatically computed values, scanned over all frames
- Colours from blue (low) to magenta (high)
- Automatic number format

These default settings are shown right, the **LEVELS** sub-menu.

Contour Level settings are stored separately for each component category. If for example the number of levels is changed to 13 and the contour levels are set to " **User def** "ined values for the "Stress" component category then these options will be used whenever a "Stress" component is plotted.

As well as being stored for the "Stress" category the new settings will also become the default for any categories that the user has not explicitly stored settings for.

If after setting the number of levels for "Stress" the user sets the number of levels for "Strain" to 10 then 10 will be used for "Strain" and for any categories that the user has not explicitly stored settings for. If the user then swaps back to "Stress" component the previously set number of levels (13) will be used.

NOTE : If the data component is set to FORMABILITY then a special set of options will be displayed instead of this menu, see Special options for FORMABILITY.

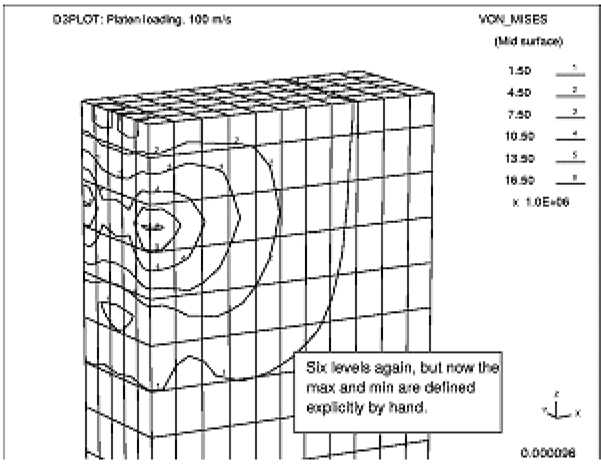
Setting contour levels (Automatic, Max_&_Min, User_Defined and Linear/Logarithmic)

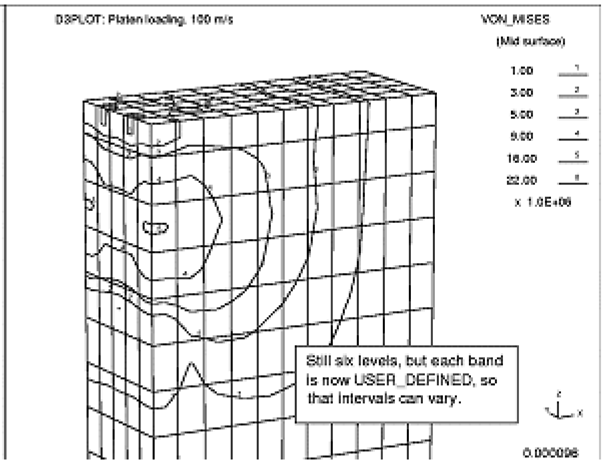
By default contour levels are **AUTOMATIC** over all frames. This means that the maximum and minimum values are computed prior to each plot, and the resulting bands spaced evenly between these.

Automatic contour bands can be computed in two ways:

Automatic mode	During a static plot	During an animation

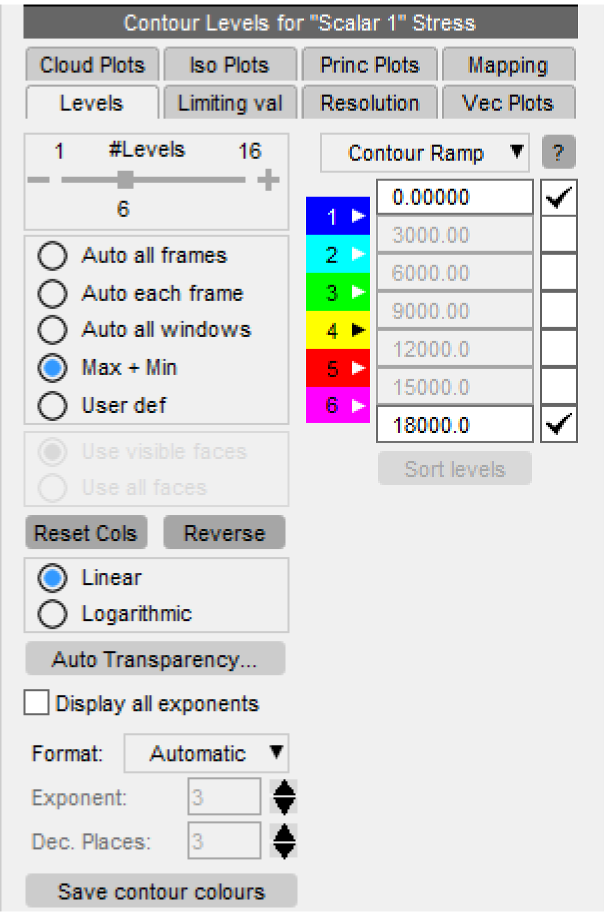
<table>
<tr><td>Over all frames</td><td rowspan="3">Contour bands are automatically scaled to the max and min values in this plot only. (The same behavior in all three modes.)</td><td>The "envelope" of max and min values of all frames making up the animation are calculated, then every frame of the animation is contoured using this same single range of values. So the contour bands and values are the same in every animation frame.</td></tr>
<tr><td>Each frame separately</td><td>The max and min values in each frame are computed separately, and each frame is auto-scaled to its own set of values. So the contour bands may change in each frame, and the effect is the same as a series of individually auto-scaled static plots.</td></tr>
<tr><td>Over all windows</td><td>The "envelope" of max and min values of all frames across all the windows in the current page making up the animation are calculated, then every frame of the animation is contoured using this same single range of values. So, the contour bands and values are the same in every animation frame across all the selected windows in the current page.</td></tr>
</table>





In the figures above the contour levels have been set manually: the max and min values only are set in the left figure, user-defined levels for each band are set in the right figure.

MAX_&_MIN levels have been selected, and the upper and lower values defined. The intermediate values are interpolated linearly and filled in for you. They can then be changed manually by enabling the tick boxes. This will switch to user-defined contours explained next.

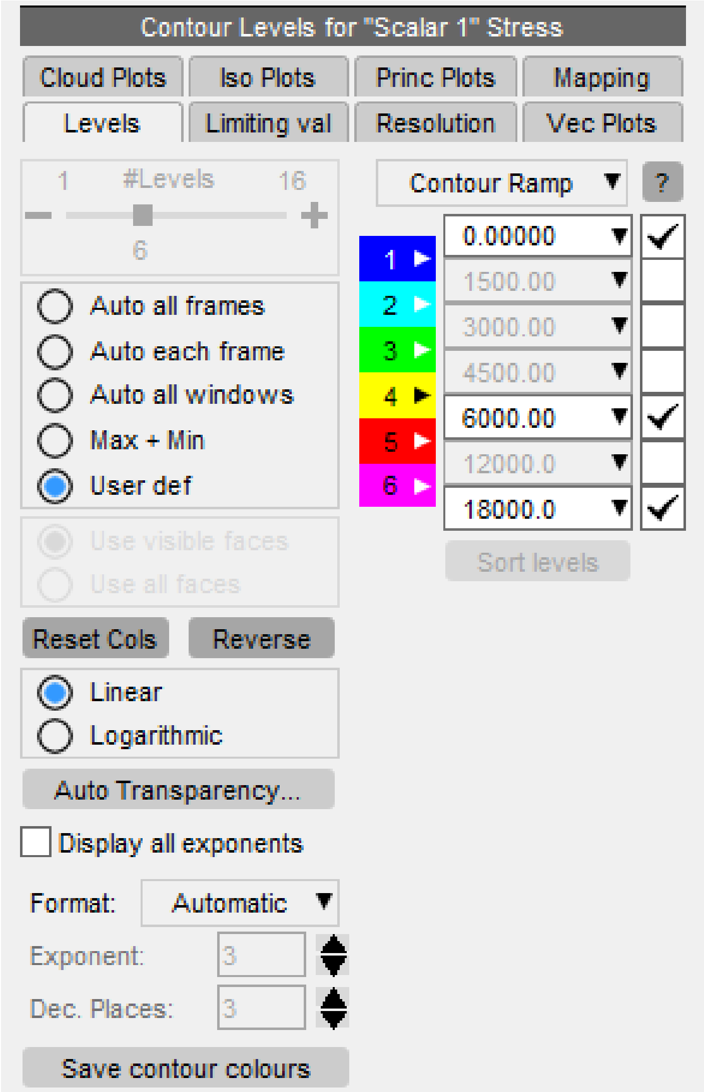


The figure (right) shows the same panel set up for the plot in (b) above. **USER_DEF** ined levels have been chosen, and apart from the minimum and maximum one user-defined level has been specified. The levels explicitly specified can have uneven intervals. When the tick boxes for levels are off, the threshold value is calculated with uniform intervals.

Also, it is possible to turn off the first and/or last level, which will then be calculated automatically like in **Auto all frames**.

New levels for user-defined contours can be inserted or existing levels deleted with the popups at the text boxes with the band threshold values.

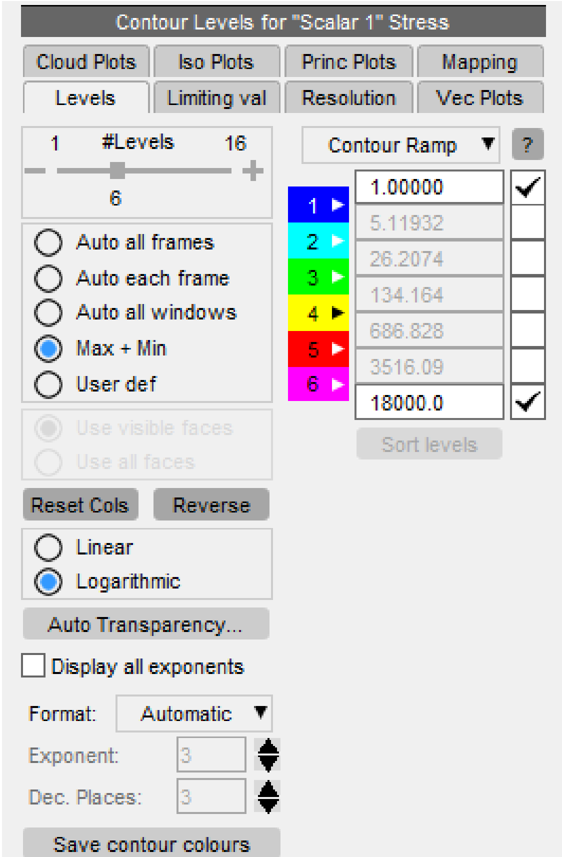
User-defined contours are only meaningful when the levels explicitly specified are in monotonic order. When this is not the case, the ticked levels can be sorted into ascending order with the **Sort levels** button.



Hint: Choose **MAX_&_MIN** first, fill in upper and lower-bound values, then turn on tick boxes for explicit levels in between. The interpolated intermediate values are preserved when turning on a level and

may give some
indication what you
might want to specify.

All levels between the minimum and the maximum will be interpolated linearly by default such that all bands have got the same width. This also applies to **User defined** levels where checkboxes are off. As alternative to that **Linear** interpolation there is **Logarithmic** interpolation, where the factor to get from one threshold to the next will always be the same. Between any two positive numbers this is the same thing as replacing them with their logarithm, interpolating linearly and reverse the logarithm.

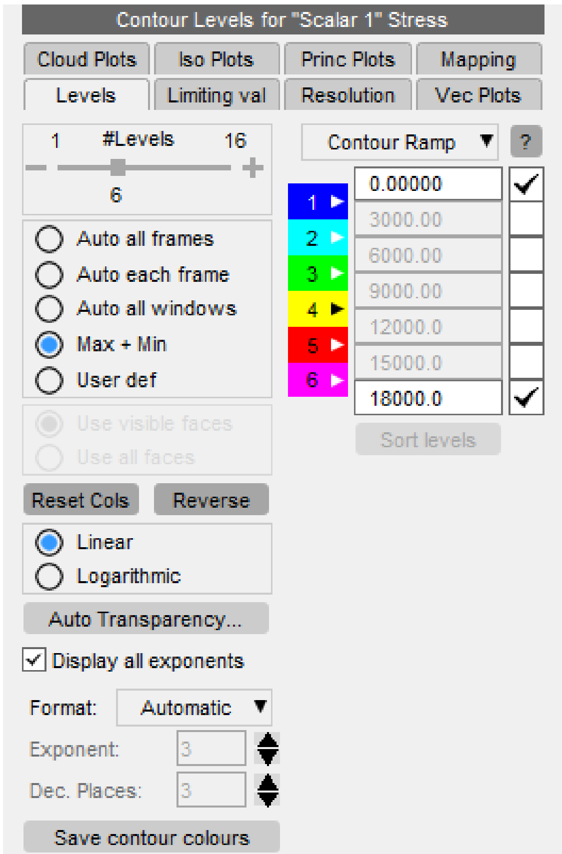


For example, logarithmic interpolation with 6 levels from 1 to 1000000 will give the log values 0 to 6, and interpolated values would be 10^0, 10^1, 10^2,... 10^6.

Logarithmic interpolation is also available between two negative numbers, for example 6 levels from -1000000 to -1 would give thresholds -10^6, -10^5, ..., -10^1, -10^0.

When logarithmic interpolation between zero and a non-zero number or between two numbers of opposite signs are requested, D3PLOT will automatically insert numbers close to zero and interpolate the negative and positive ranges logarithmically separately. By default it will divide the extremal thresholds by 2 recursively, but other factors can be obtained by switching to user defined contour levels where thresholds close to zero are specified explicitly with checkboxes turned on.

The **DISPLAY ALL EXPONENTS** switch will put an exponent on each contour bar value rather than one exponent at the bottom of the bar that applies to all values. This is useful if the scale has been converted to a log scale so that the individual values can be shown with enough precision.



Computing contour bands over multiple windows and models

Where there is more than one window, possibly showing a different component, on a model; or more than one model in the database then contour levels are computed as follows:

AUTOMATIC contours:

For each active window (**Wn** tab selected):

- The max and min values of all visible elements in all models is computed
- These become the max/min bounds for that window
- Changing what is displayed in that window will update these bounds
- Each window is independently calculated, regardless of the contents of other windows for auto all frames and auto each frame contour modes.
- During animation exactly the same rules apply, except that the "envelope" of values from all frames is used to calculate the max & min values.

MAX & MIN contours:	For each active window (`Wn` tab selected): • The user-defined max and min values are applied to that window, regardless of contents.

Applying the same contour bands to all windows

It is often the case that you have several windows, and you want to have the same contour bands in all of them. To do this:

If you want to specify the bands to be used:	• Make sure that the `Wn` tabs for all required windows are selected. • Choose **MAX_&_MIN** levels • Set the required levels.
If you want to find the envelope of all windows, derive a max & min value from that, then set it in all windows.	• Make sure that the `Wn` tabs for all required windows are selected. • Choose **AUTOMATIC** levels. • Perform plots in all windows to update their local bands. • If you want values over an animation then animate all these windows to force computation of the "envelope" max/min values in each. • Then select **MAX_&_MIN**

The second method works because when you switch from **AUTOMATIC** to **MAX_&_MIN** D3PLOT computes the "envelope" of max/min values from all active windows, and applies this as the default values for **MAX_&_MIN** mode.

It is important to understand the distinction between automatic contouring over animation ***frames*** , and automatic contouring at a given ***state*** .

Automatic contour bands during Animation

Case 1: When Automatic all frames mode is in use.

During an ***animation*** , or when a particular ***frame*** of an animation is displayed statically:

- For each window D3PLOT will scan all selected states and find the max and min contour values.
- If the window contains more than one model the max and min over all models in the window is found.

These max and min values become the contour bounds used for the animation, or when any frame of the animation is displayed, with the intention that contour bands in any frame will have the same values.

Case 2: When Automatic each frame mode is in use.

Each frame of the animation is auto-scaled separately and whenever a frame is display, either statically or when animating, the contour bands will be "local" to that frame.

In this mode the contour bands will usually change during each frame of an animation, therefore the values assigned to a particular contour band will not normally be the same in successive frames.

Case 3: When Automatic all windows mode is in use.

During an ***animation*** , or when a particular ***frame*** of an animation is displayed statically:

- For each window D3PLOT will scan all selected states and find the max and min contour values.
- If the window contains more than one model the max and min over all models in the window is found.

These max and min values across all frames and windows in the current page become the contour bounds used for the animation, or when any frame of the animation is displayed, with the intention that contour bands in any frame will have the same values.

Commands which display animation ***frames*** are:

- [**>**] Play to initiate an animation, either at the top of a graphics window or in the "States" panel.
- Any of the **<< |< >| >>** frame positioning commands either at the top of a graphics window or in the "States" panel.
- Using the state slider at the top of a graphics window.
- Using the <shift> + <arrow key> short cut to toggle through frames.

Automatic contour bands during static state display

During static display of a specific ***state*** :

- For each window D3PLOT will find the max and min of all models in that window at that state only.

If you subsequently move on to another state then the contour bounds will change as the new max and min values for that state are used instead.

Commands which display static ***states*** are:

- Any explicit data plotting command (eg **SI** , **CT** , ..) while not animating.
- Using the state slider in the "States" panel.
- Setting an explicit state number or time in the "States" panel.
- Using the <arrow keys> (no shift) to toggle through states.

A useful trick if you want to "animate" a series of states, but to auto scale contours to each state individually, is to use the <arrow keys>: hold them down letting them auto-repeat and D3PLOT will cycle through states with specific contour bands for each state.

Why is there a distinction between "frames" and "states"?

It is true that for most transient analyses "frames" will be equivalent to "states", however there are some cases where this is not the case:

- When plots are interpolated by time the frame vs state equivalence no longer holds, and typically there are many more frames.
- The user can choose to animate only a subset of the available states, reducing the number of frames.
- In frequency domain (modeshape) analyses each "state" is a mode, and animation frames cycle through +/- 180 degrees at that mode.
- Similarly "static" analyses with a series of loadcases use frames to animate each case in a quasi-modal fashion.

Preserving this distinction makes D3PLOT more flexible and provides more options for contouring animations.

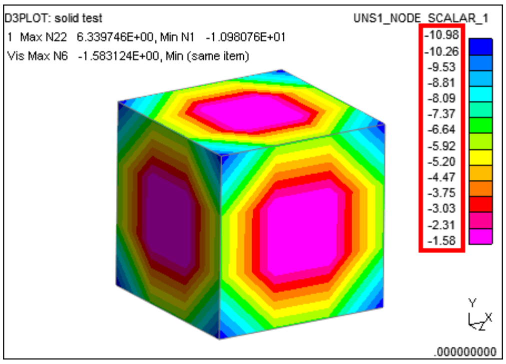
For more details about animation, frames and states, and how to specify them, see [Animation How to Display, Control, Store and Retrieve Animation Sequences]() on Animation.

Visible faces vs All faces

By default D3PLOT only uses the values on the visible faces of 3D elements when using AUTO or AUTO EACH to calculate the contour bar limits for SI, CT or LC plots.

If an internal face has a higher or lower value then it is reported in the top left of the window but the value is not used for the contour bar range.

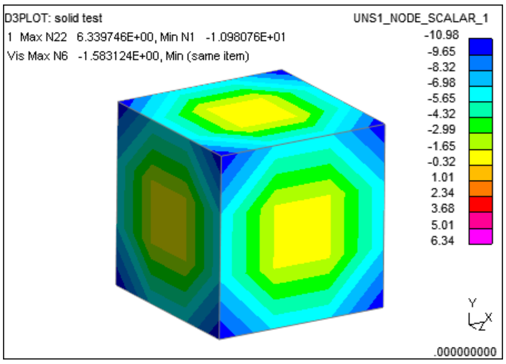
(If internal faces are turned on then the contour bar includes the values)



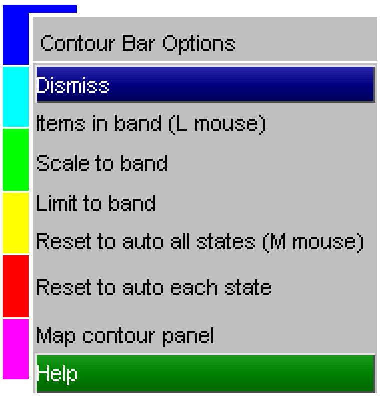
If this option is changed to **Use All Faces** D3PLOT use the internal face values when calculating the contour bar range.

This option can be set as the default in D3PLOT via the preference option.

d3plot*contour_bar_3d_faces:



Clicking on the contour bar to set levels



As an alternative to the explicit methods of setting contour bands described above it is also possible to set and restrict values by clicking on the contour band display itself.

On a data-bearing plot hover the cursor over the contour level bands, and the cursor symbol will change to `CONT/OPTS` , and the following options are then available:

- Left mouse: limits the display to only those items within the selected band
- Middle mouse: reverts to automatic contour levels
- Right mouse: maps the options menu shown here

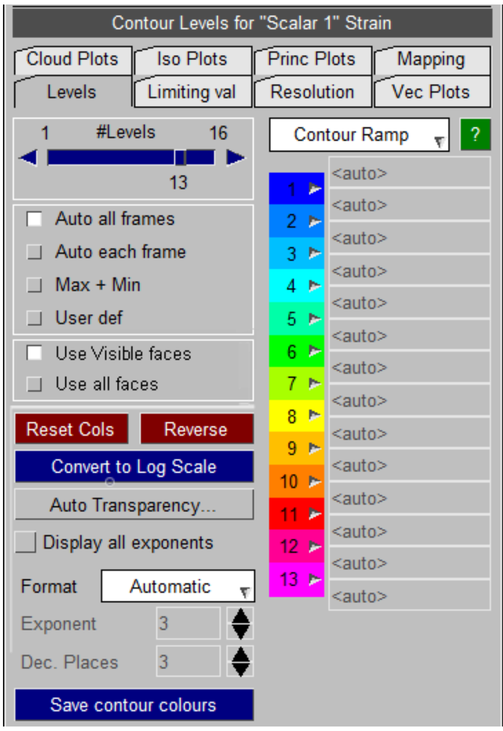
`Items in band (left mouse)`	Restricts the display to those items in the selected band, but does not alter the overall contour band limits. The effect is based on the centre value of elements, so if contouring is on (the default) you may see gradations of value outside the limits of the band chosen. To prevent this turn averaging off, or select a plotting mode (such as **CL** oud plots) which shows centre values only.
`Scale to band`	Resets contouring to max/min using the upper and lower values of the band chosen. Display is not limited to this range, so items outside the range will still be drawn.
`Limit to band`	As `Scale to band` , but also sets limiting values to the original band's max/min, so only elements within the original band are shown.
`Reset to auto all states (middle mouse)`	Resets contouring to **Automatic** (all states), and turns off limiting values if switched on by one of the options above.
`Reset to auto each state`	Resets contouring to **Automatic** (each states), and turns off limiting values if switched on by one of the options above.
`Map contour panel`	Is the equivalent of selecting **Contour > Levels** .

Restriction to a band is carried out using the **Contours > Limiting Values** function described in LIMITING_VALUES... Limiting What is Contoured by Value Range . In effect the functions here set the upper and lower bounds, and turn on limiting. You can adjust these further by hand if you wish.

Changing contour band colours

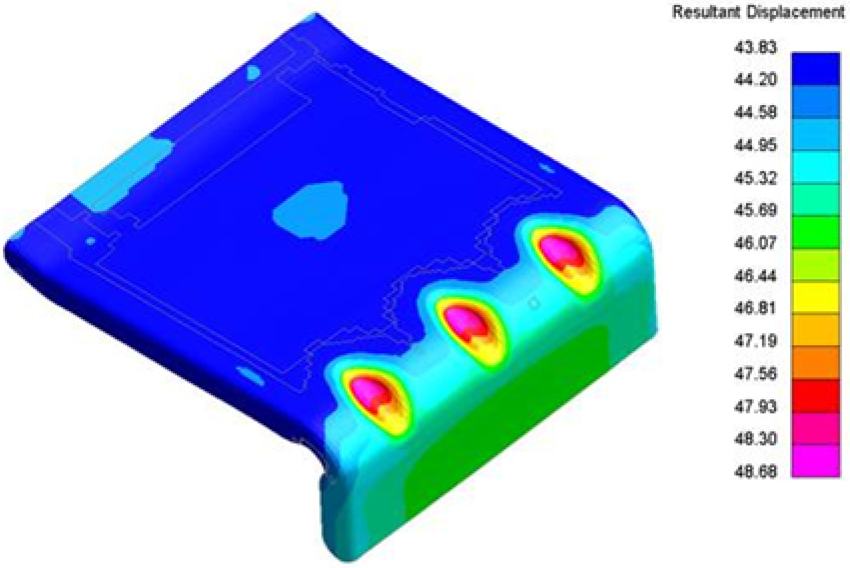
By default colours range from **blue** (low value) to **magenta** (high value), and the colour range is set up automatically. Internally there are 15 standard colours to choose from, and any contour band may be assigned any colour.

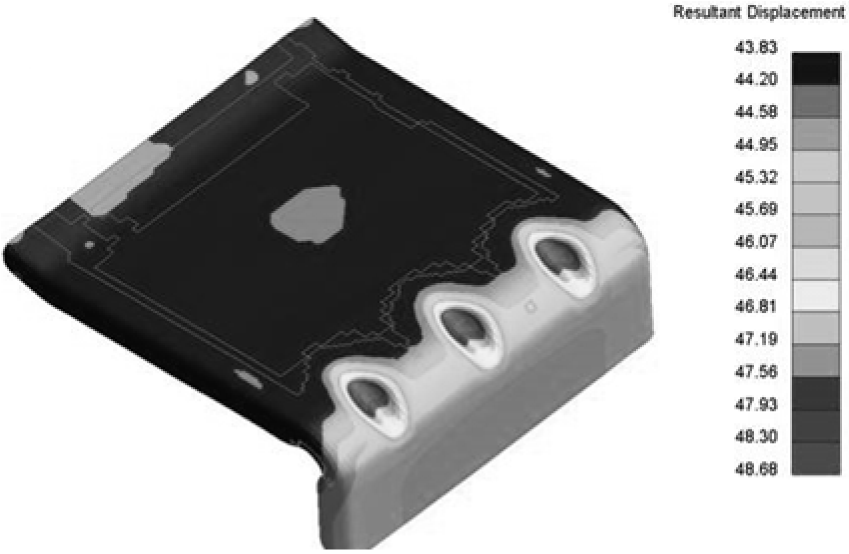
To change a colour click on its number then select an alternative standard colour from the panel, or define your own arbitrary colour.



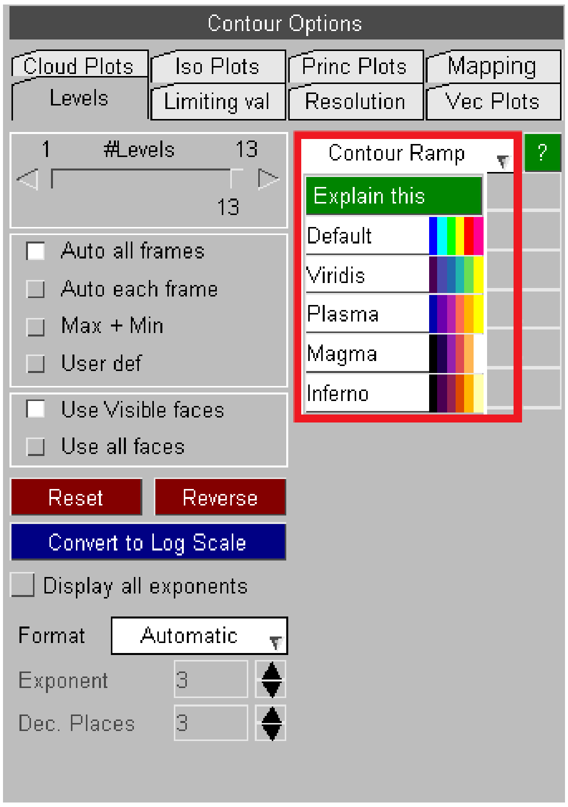
SAVE CONTOUR COLOURS	Will save the contour colours to the oa_pref file so they will be reloaded when a new session of D3PLOT is started. This can be useful if you regularly modify the colours, e.g. setting the minimum value to grey.
RESET	Will reset contour colours to their default range for this number of levels.
REVERSE	Will reverse the current colour range for this number of levels.

The default colours are not colour-blind friendly and when printed in greyscale, for example in a report, it is not possible to distinguish the different contour bands.

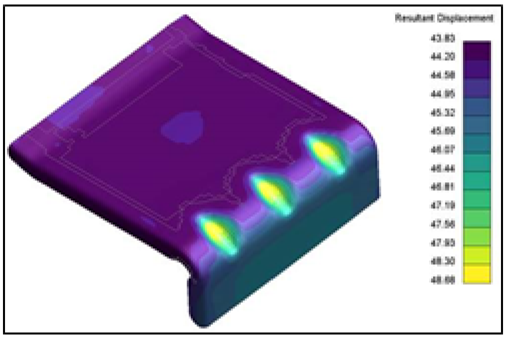

Resultant Displacement
43.83
44.20
44.58
44.95
45.32
45.69
46.07
46.44
46.81
47.19
47.56
47.93
48.30
48.68


Resultant Displacement
43.83
44.20
44.58
44.95
45.32
45.69
46.07
46.44
46.81
47.19
47.56
47.93
48.30
48.68

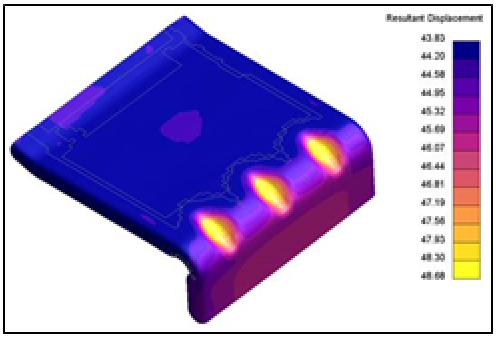
To address this there are some new contour ramps which can be selected from the popup above the contour ramp colours. These are colour-blind friendly and when printed in greyscale each contour band is distinguishable from the others.



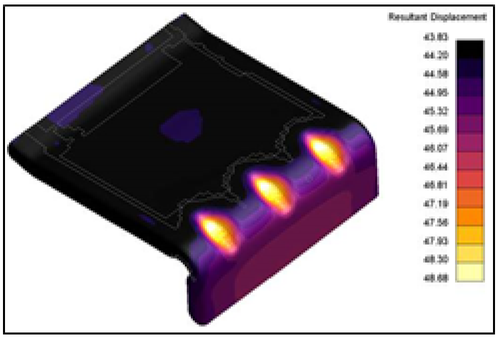
Viridis



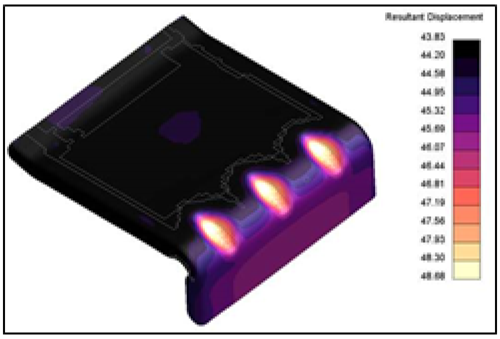
Plasma

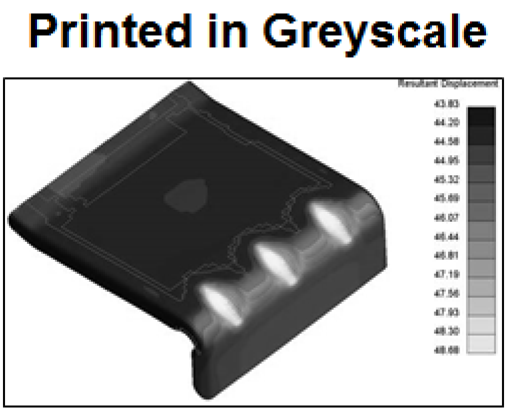


Inferno



Magma





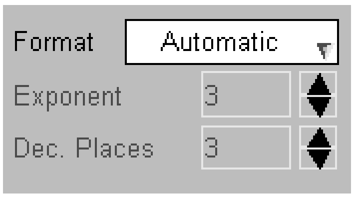
Note 1: Contour bands define the upper and lower values of each discrete band. For solid contoured plots (i.e. **CT**, **SI**) each band lies between these limits. For line contoured **(LC)** plots each line will lie at the mid-point of its band.

Note 2: Colour tables for contours are stored separately for each number of contour levels. So if you change colours for (say) 6 contour levels this will not affect colours for any other number of levels.

Note 3: During **SI** shaded-image plots the current number of contour levels is mapped onto 21 colour bands, interpolating linearly, regardless of the actual number of bands selected. This is to improve the colour resolution of plots. As a consequence colours are also interpolated within these 21 bands from the **#levels** set here. Thus defining more contour levels will give finer control over the colours used in shaded-image plots.

Note 4: Whichever way they are defined, contour bands must be in ascending, monotonic order. This is particularly significant for **USER_DEFINED** mode: you will not be permitted to create bands that have zero or negative intervals.

Number format : Controlling the number format of contour values

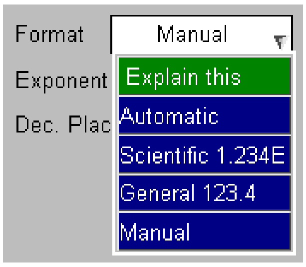


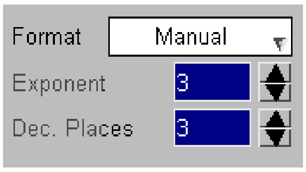
By default D3PLOT tries to work out a sensible number format to display the contour values in. In some cases the user may want to change the default behavior and this can be achieved here.

You can select either 'Scientific', 'General' or 'Manual' to control how the numbers are formatted.

The number of decimal places used can be set for 'Scientific' and 'General' formats.

For the 'Manual' format both the number of decimal places and the exponent value can be set.





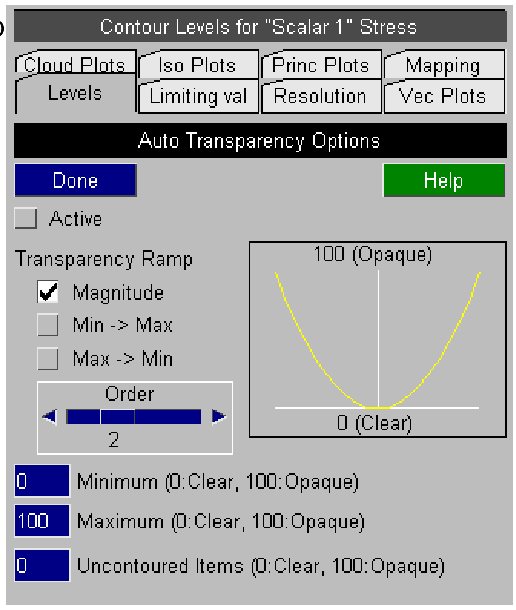
Automatic Transparency

The Automatic Transparency option can be used to automatically set the transparency of entities based on the data values being plotted on them.

In complex plots it is often difficult to locate minimum and maximum values as they are often hidden behind other entities. Although entities can be blanked to reveal these "hidden" parts this is sometimes unsatisfactory as a solution.

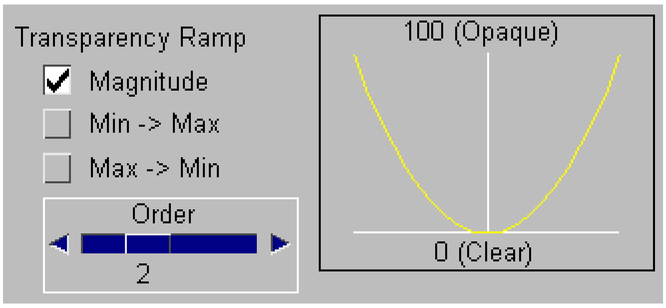
By varying the transparency of entities it is possible to view these minimum and maximum values while still viewing the whole structure.

This option affects the transparency of entities in SI, CT and CL plots.

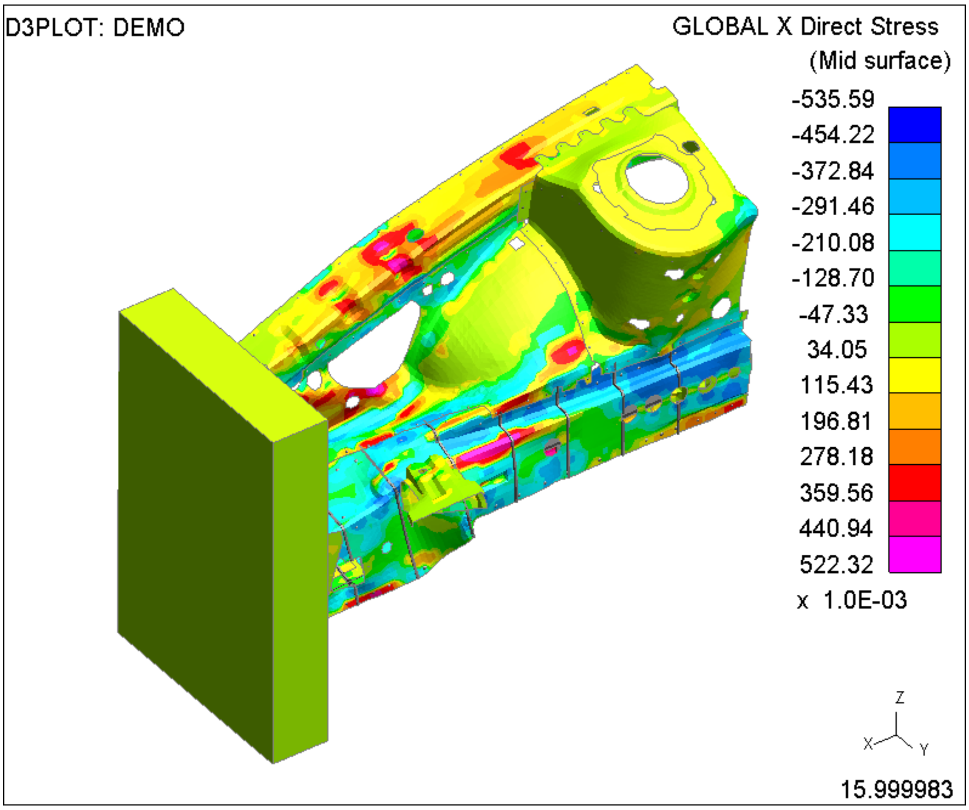


Transparency Ramp

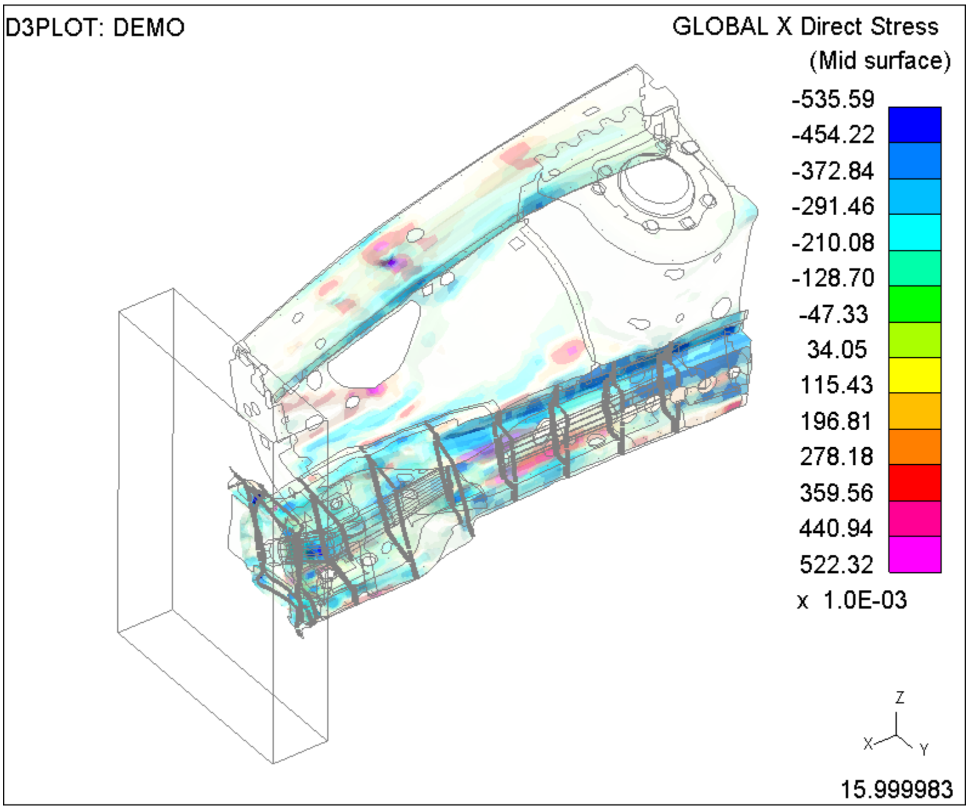
These options control how the transparency levels are calculated for the data values being plotted.



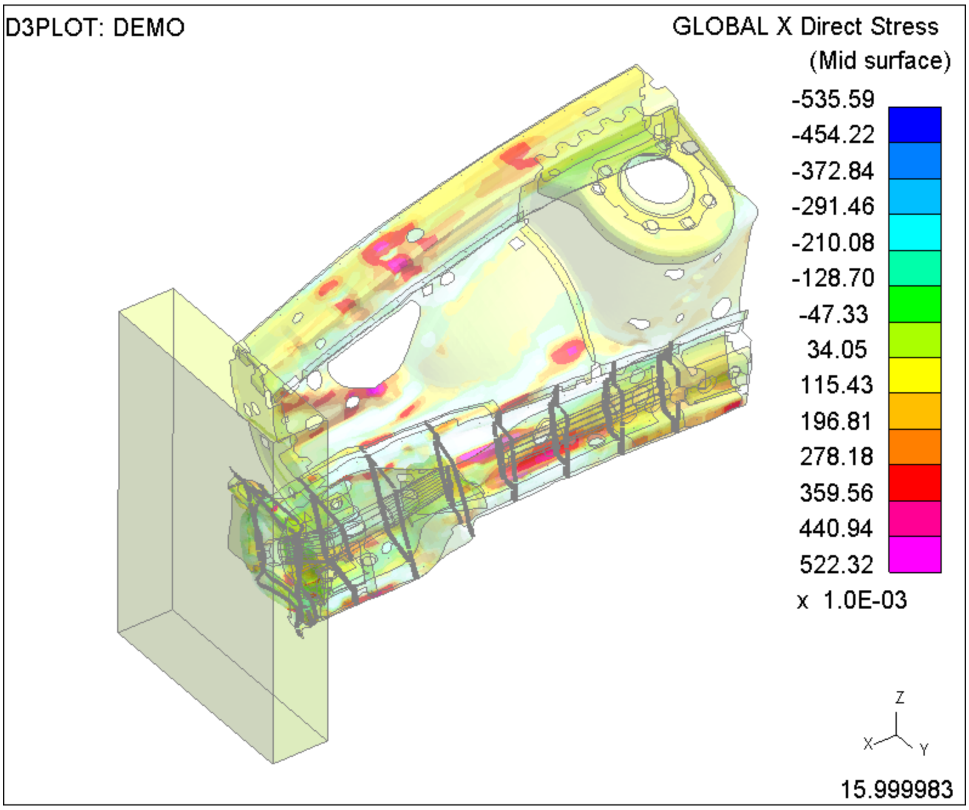
Magnitude	Transparency of entities is scaled by the magnitude of the data values.
Min -> Max	Transparency is scaled from the minimum data value (clear) to the maximum (opaque).
Max -> Min	Transparency is scaled from the maximum data value (clear) to the minimum (opaque).
Order	This option controls the shape of the transparency ramp. Order 1 generates a linear ramp using the data value (X), order 2 generates a ramp based on (X^2), order 3 uses (X^3) and order 4 used (X^4)



Default SI plot

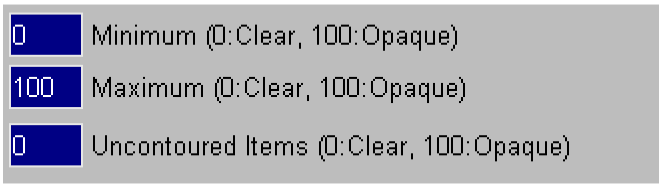


SI plot with default automatic transparency options



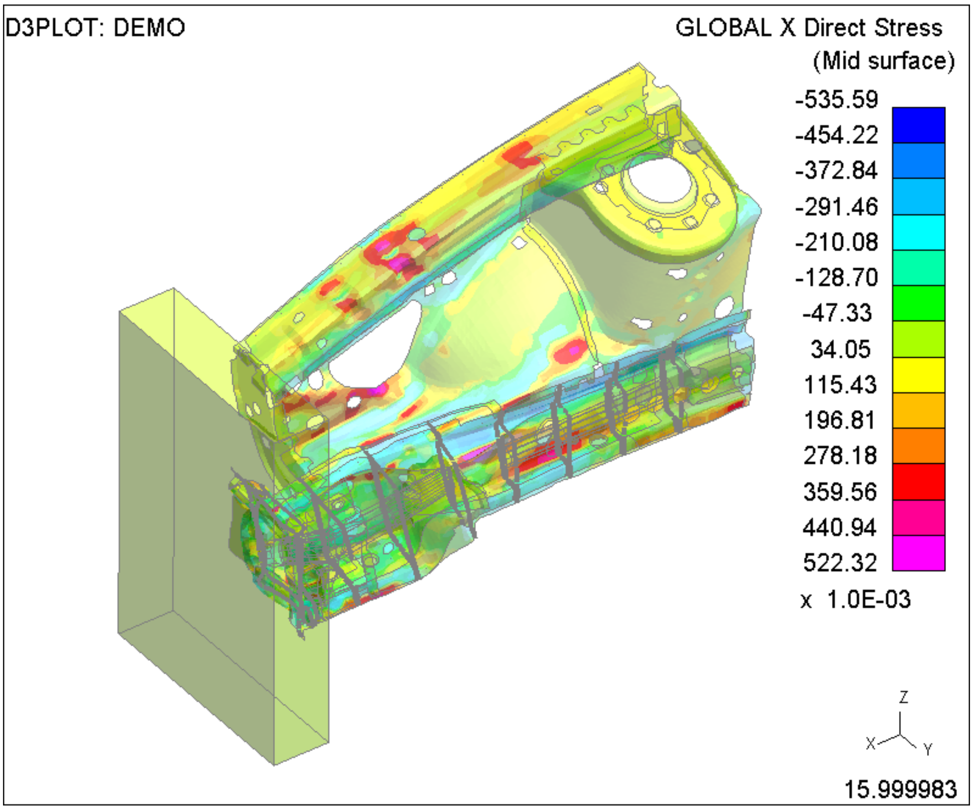
SI plot with transparency scaled from Min ->Max

These options control the minimum and maximum transparency values that are used when calculating the transparency ramp.

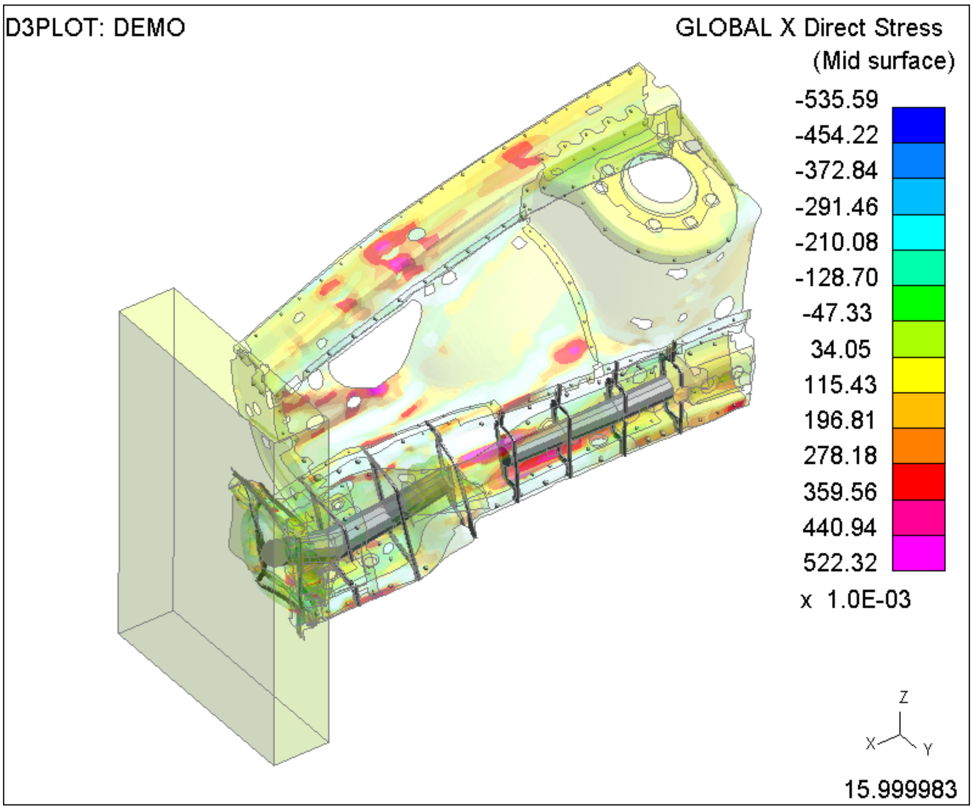


The values must lie in the range 0 (clear) to 100 (opaque).

A separate transparency value can also be defined that is used for all items that are not being contoured.



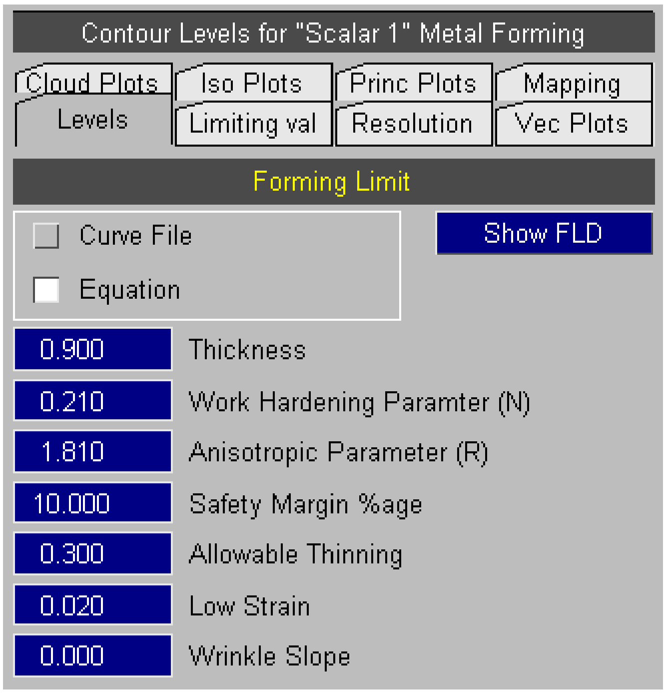
SI plot with transparency scaled from Min ->Max and a minimum transparency value of 30



SI plot with transparency scaled from Min ->Ma and uncontoured items set to 100 (opaque)

Special options for FORMABILITY

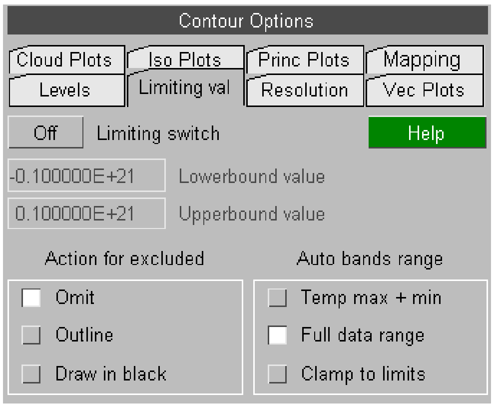
If the data component is set to FORMABILITY (in the Metal Forming category) then the Contour Levels menu is replaced with the options opposite, for more details see METAL FORMING.



LIMITING_VALUES... Limiting what is contoured by value range

By default all elements are contoured, regardless of whether or not they lie within the max & min range of the contour bands specified. Elements lying outside these bands are given the highest or lowest contour band colour as appropriate.

In some cases you may wish to contour only those elements which lie within a restricted range of values, and to omit (or draw only in outline) those outside this range. This can be done with the LIMITING_VALUES option.



This figure shows the control panel for the LIMITING_VALUES options.

When the Limiting switch is off, (default), the other options in this panel are greyed out.

Lowerbound value is the value below which data are not contoured.

Upperbound value is the value above which data are not contoured.

Action for excluded defines what happens to excluded elements, see below.

You need to consider what action is to be taken with elements that are not contoured. There are three options, which it is convenient to take in reverse order.

The `Auto bands range` options determine how Limiting Values interact with automatic contour bands when limiting is active:

`Temp max + min`	Is used when the short cut "click on a contour band" option in [Clicking on the contour bar to set levels]() has been used. This temporarily sets

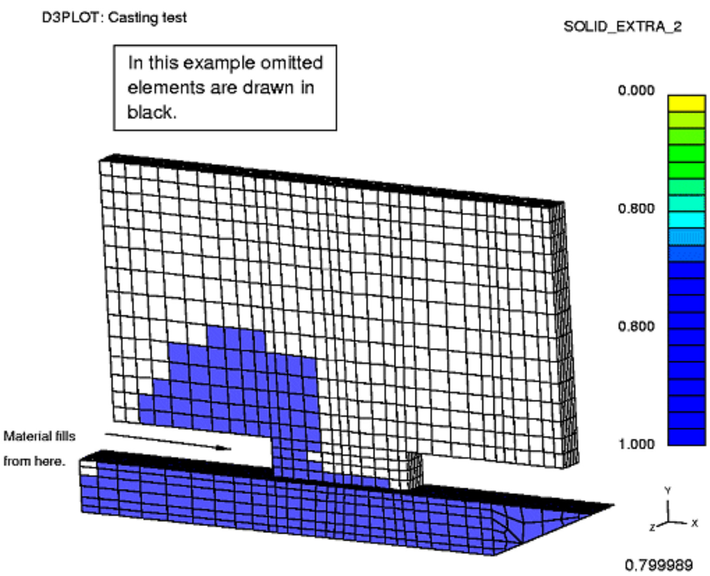
	contouring to enforced max/min values, which conflicts with this mode, and should not be set manually.
`Full data range`	If automatic contour bands are in use then they will adjust themselves to the full range of data available, ignoring the lower and upperbound values set here. Therefore the contour band boundaries will not change as the upper and lowerbound values are changed here.
`Clamp to limits`	Restricts automatic contour bands to the limits set here, changing as they change. Therefore whatever is drawn on the plot will exploit the full range of contour bands and colours available, making this suitable if you want the maximum colour and band discrimination for what is drawn.

The following three examples are a mold filling (fluids) analysis in which a plate comprised of solid elements is filled from below. They show the effects of the three Action for excluded options, in particular note how this affects the display of internal structure:

This figure uses DRAW_IN_BLACK for elements that lie outside the limiting values. (Black is reproduced as white on a hardcopy as here.)

Here we are plotting "void fraction", ie %age fullness of fluid, and only values in the range 0.8 to 1.0 (ie 80% full or more) are shown.

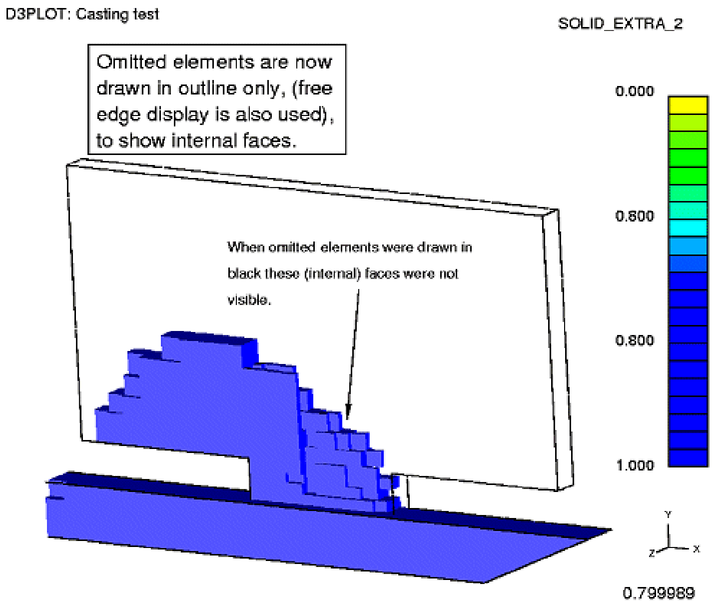
It is clear that no internal structure is visible since elements on the outside faces, which are not full, obscure it.



This figure uses OUTLINE for elements that lie outside the limiting values.

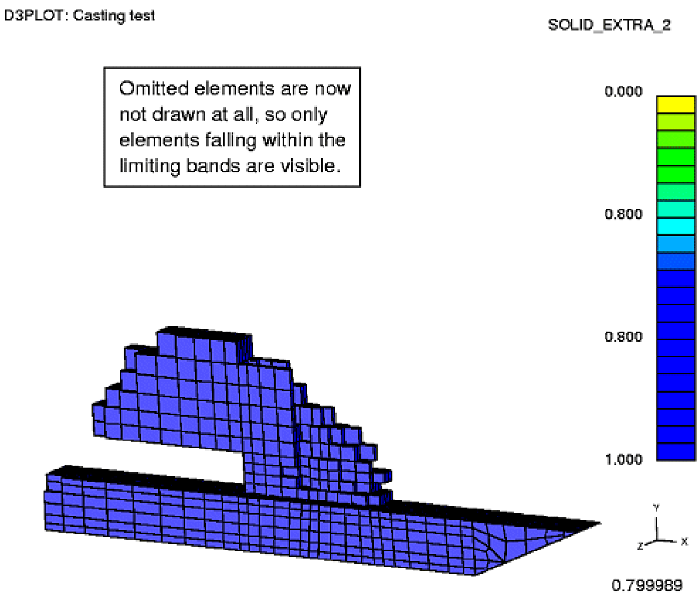
Only the outline of elements which are out of range is drawn, so that they do not obscure structure behind them.

In this example free edge display ([DISPLAY_ OPTIONS] FREE_EDGE) has also been used to remove clutter of excess mesh.



This figure uses OMIT for elements that lie outside the limiting values.

Now only those elements that are within the specified limits are drawn at all, those outside them are effectively blanked from the display.



The default state of the LIMITING_VALUES switch is off, so that none of these settings apply.

When turned on it will apply to any data plotting mode, but clearly the OUTLINE and DRAW_ IN_BLACK options for excluded elements are only meaningful for 2D and 3D elements.

7.4.2.3. RESOLUTION... Setting Contour Resolution

RESOLUTION... Setting contour resolution

Structural elements

The output from most 2D & 3D elements in LS-DYNA is only written at element centres, implying constant stress: no values are written at nodes on elements. Therefore contouring, by displaying variations of data across elements, is an approximation used to help visualization of the data distribution.

For fully integrated SHELL elements D3PLOT can use the data to more accurately contour the variation of data across the element (provided MAXINT on the *DATABASE_EXTENT_BINARY card is negative and a ZTF file has been written from PRIMER).

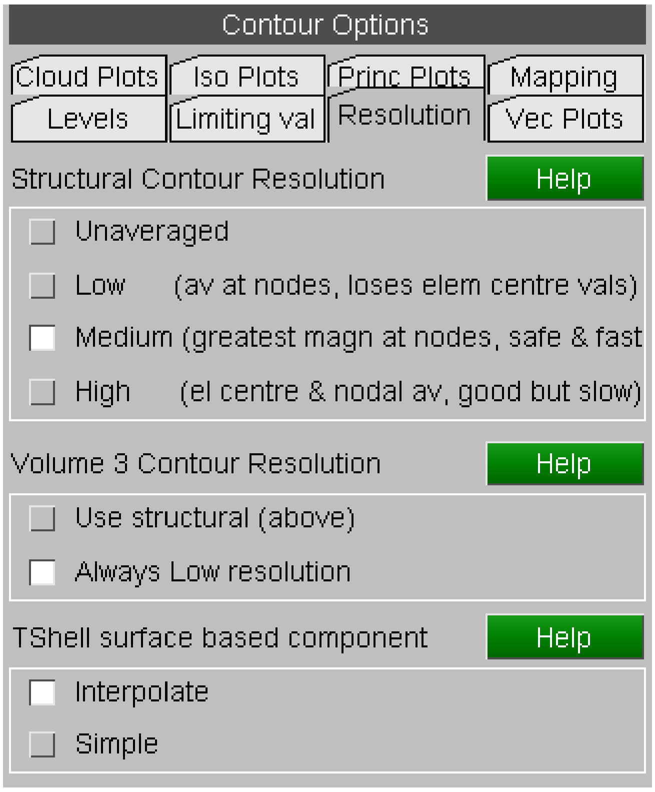
Most LS-DYNA elements are constant stress, and experience has shown that the safest - if not necessarily the most accurate - method of averaging data at nodes for contouring is to use the largest magnitude of all the element results meeting at the node, the "Medium" resolution option. This tends to over-estimate the geographical extent of peak values, but this is preferable to losing these peaks by averaging data at nodes. However other more traditional methods are available, and are described Contour averaging options for structural elements.

Volume 3 elements (ICFD, CESE, EMAG, etc)

These analyses vary: some solution methods write results at element centres and others at nodes. However experience suggests that these analysis types work best with some degree of data smoothing, so it is possible to set their contour averaging method independently to "Low" resolution where results are averaged at nodes.

Thick shell contour for surface-based data

Thick shells are a special case: they use explicit nodes to define their thickness and hence are drawn as pseudo-3d elements with distinct top, bottom and sides. D3PLOT can display surface-based (stress, strain and extra) data either "simply" for a single surface on all faces of the element, just like thin shells. Alternatively it can show top surface data on the top face, bottom surface data on the bottom face, and interpolate results on the side faces. This is described in more detail Contour Display Options for Thick Shell elements.



Contour averaging options for structural elements

Three levels of contour resolution are provided for structural elements

LOW	Data are averaged at nodes, then contour bands (or lines) are drawn linearly from edge to edge across elements. The element centre values are not directly included so some smoothing of results occurs and peak values may be lost: which is potentially unsafe.

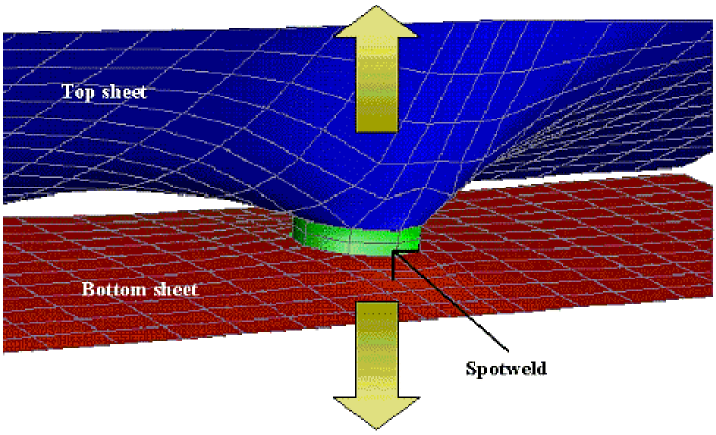
MED IUM	The value with the greatest magnitude at each node is found, then contour bands (or lines) are drawn linearly from edge to edge across elements. The element centre values are implicitly included since they will qualify as maxima at nodes: a safe overestimate.
HIGH	Data are averaged at nodes as before, but elements are then split into sub-areas using centre and mid-side values. This enables variations across elements to be seen, and peak centre values included. However it requires up to eight times as much computation, graphics storage and drawing effort as the other two modes.

The default mode is **MED** (ium) resolution since this is both "safe", (peak centre values are included, albeit smeared out to element verticies), and quick (computation, graphics data storage and drawing effort are small).

To understand the effects of the three possible contour settings consider the following example:

Two sheets are spot-welded together, and are then pulled apart.

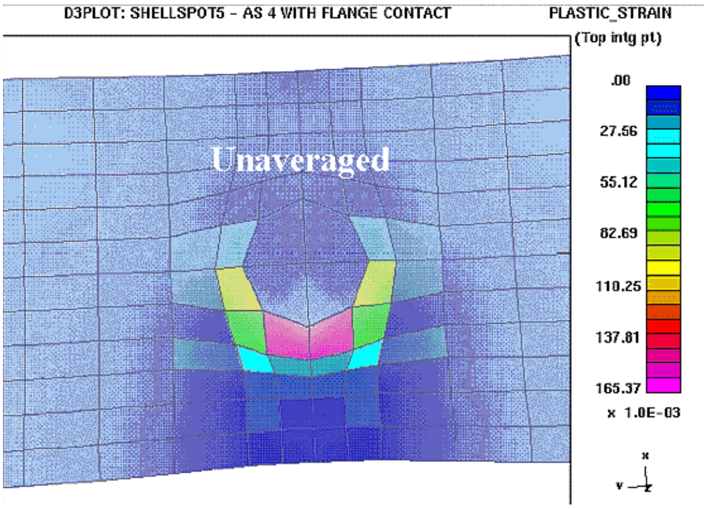
The four plots below show the results plotted as unaveraged (true) data, and also using the three data averaging methods described above.



Unaveraged data

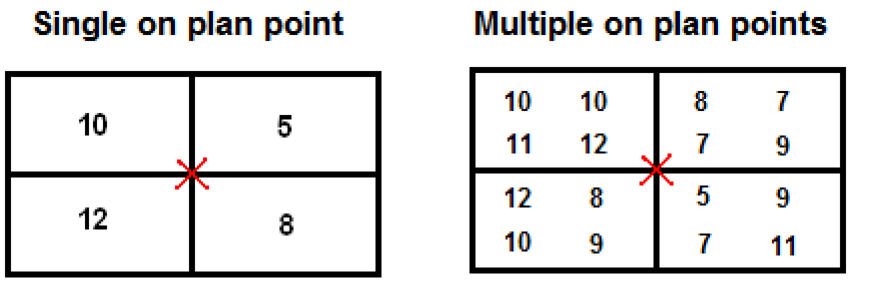
This shows the actual "true" results that come from LS-DYNA.

It is the most accurate, if not the mostly visually appealing, way of showing results in constant stress elements. However it is by definition not "contouring".



LOW resolution contours

In this mode the results from all elements meeting at a node are averaged to give a nodal value, and then these nodal values are contoured.



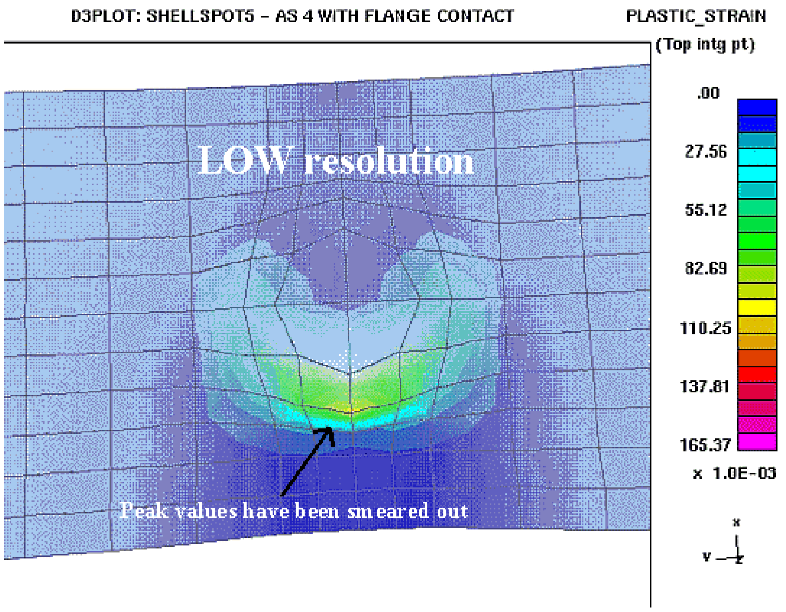
Result at centre node is the linear average:

(10 + 5 + 12 + 8) / 4 = 8.75

Results at centre node is the linear average of the closest on plan integration point on each connected shell:

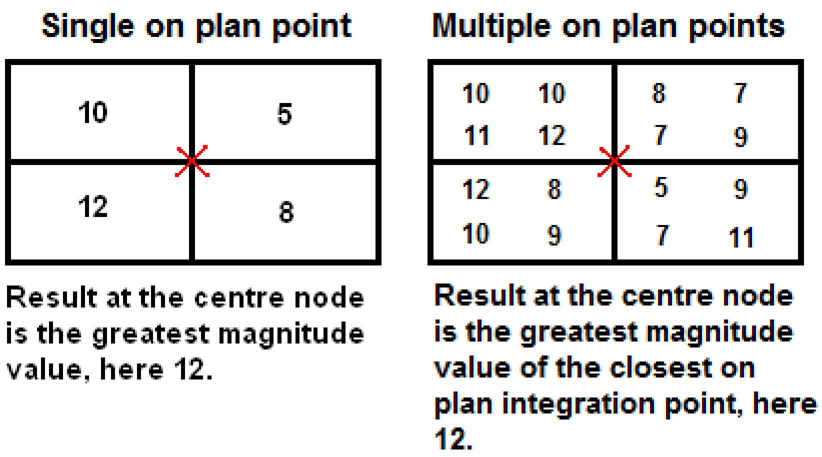
(12 + 7 + 8 + 5) / 4 = 8.00

Note that this method ***always reduces peak values*** . If you are interested in the worst case values then you should not use low resolution contours, as they are not intrinsically safe, however they are suitable for smoothing out noisy values.



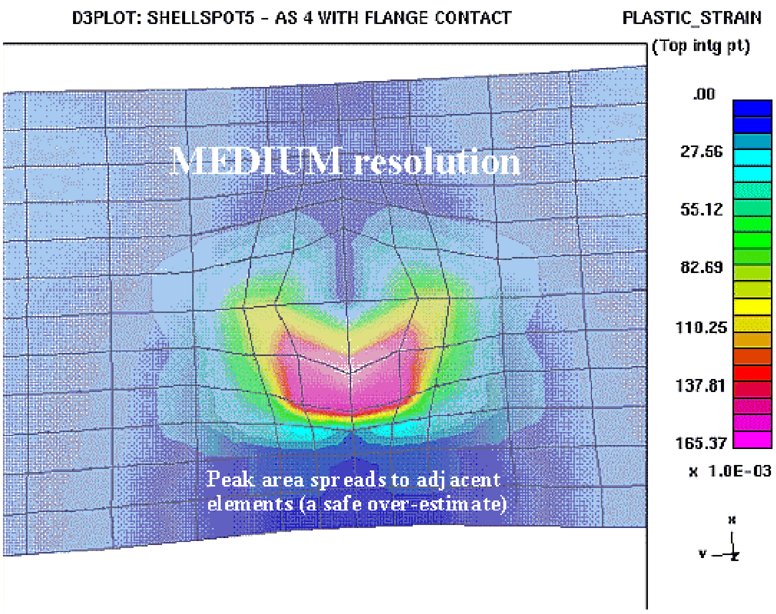
MEDium resolution

In this method results are not averaged, rather the element value with the greatest magnitude at a given node is used. These values are then contoured.



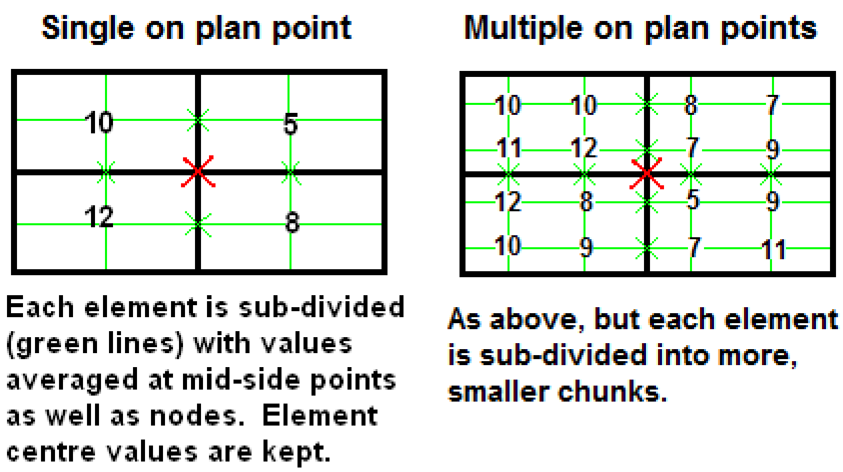
This method ***over-estimates the geographical extent*** of the peak value, here you can see that the purple extends into adjoining elements, but it is safe since the worst case values will not be lost due to averaging.

The method is the default in D3PLOT for constant-stress structural elements since it is fast to compute and draw, and also safe.



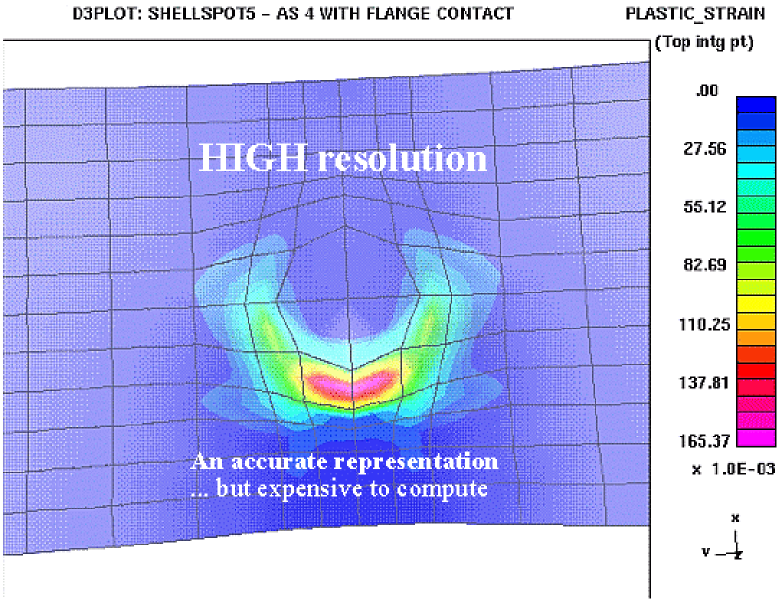
HIGH resolution contours

In this method results are averaged linearly not only at nodes, but also at element mid-side points, which allows elements to be split up into [centre], [mid-side point], [node], [mid-side] sub-elements. Contouring is then performed on these sub-elements.



This method is both ***safe and accurate*** . It does not lose element centre values and gives the most realistic display of data variation.

However it is slow to compute (for single on plan points: 9 locations per quad instead of 4, for multiple on plan points: 16 locations) and also slow to draw (for single on plan points: 4 sub-elements per quad instead of a single element, for multiple on plan points: 9 sub-elements), which is why it is not the default.



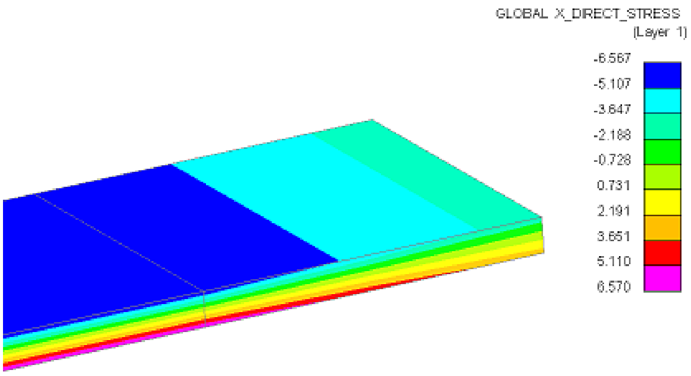
Contour Display Options for Thick Shell elements

This only affects how per-surface data is displayed on the element faces. It has no bearing on the low / medium / high resolution averaging methods described above.

Interpolate case

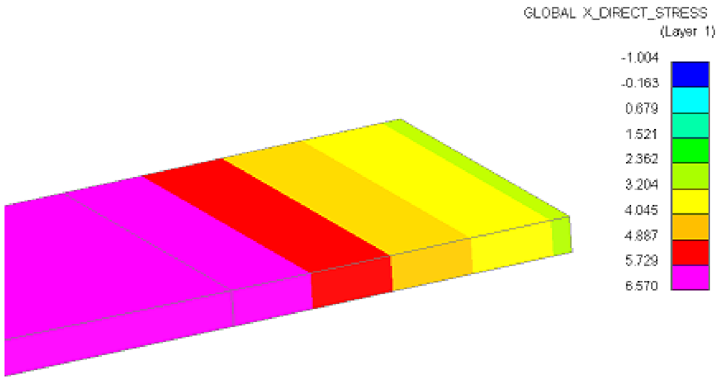
In this case the data displayed on the top surface of the shell is from the topmost (outer) integration point, and that on the bottom surface is from the bottommost (inner) integration point.

Data on the side faces is interpolated.



Simple case

In this case data for the same surface, here layer 1, is displayed on all faces of the element.



7.4.2.4. Vec Plots

Vec Plots

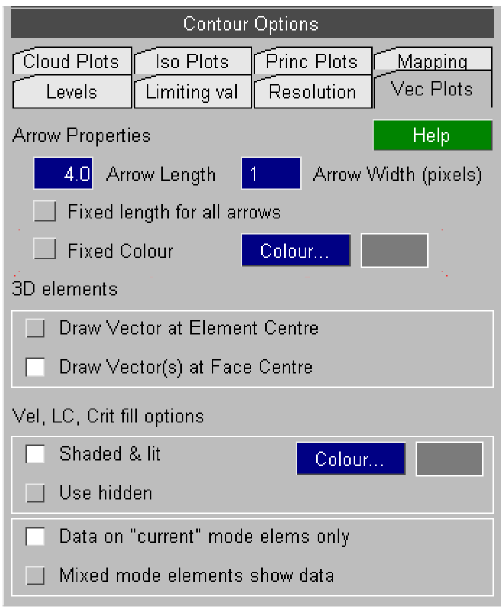
VEL velocity plots, and VEC vector plots of data, both use arrows to show the direction and magnitude of the data.

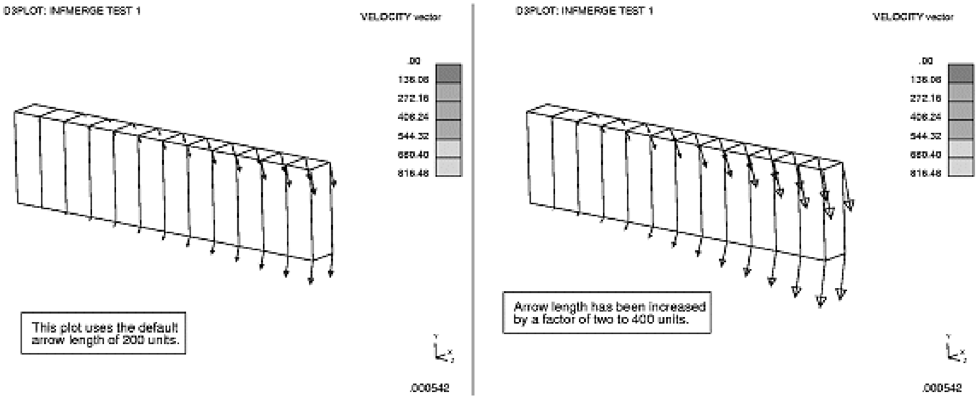
The maximum arrow length is set by default based on the longest model dimension and other values are scaled in proportion.

The figures below show examples of a velocity plot with default and double length arrows.

Vel, LC, Crit Fill Options

These options controls the appearance of element in the plotting modes where data vectors are imposed on top of "structure". For more information see [Vel, LC, Crit fill options]()





7.4.2.5. Cloud Plots: Controlling the Attributes of CL Mode Plots

Cloud Plots: Controlling the attributes of CL mode plots

By default "Cloud" plots of element-derived data show the unaveraged element value at the element centre. Selecting "Averaged at nodes" causes element-derived data to be averaged at nodes and displayed at node locations.

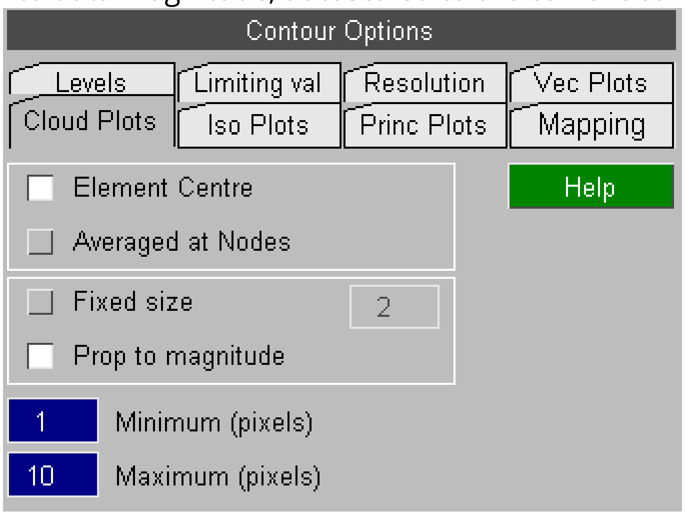
Note that:

- Averaging element data at nodes effectively gives low resolution contouring, meaning that any peak values at element centres may be "smeared out" giving lower over all maximum values.

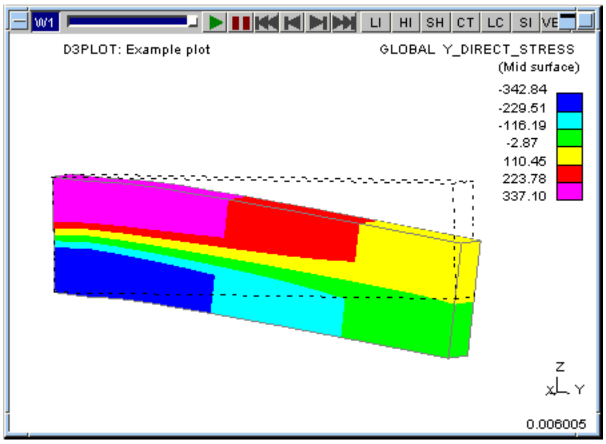
- Nodes have no unique "parent" element, so data may get averaged across parts or dissimilar element types. The results shown will be the same as those obtained from WRITE and XY_DATA for element data at nodes. (Nodally-derived data is unaffected by this setting, and is always displayed at node locations.

Point size may be:

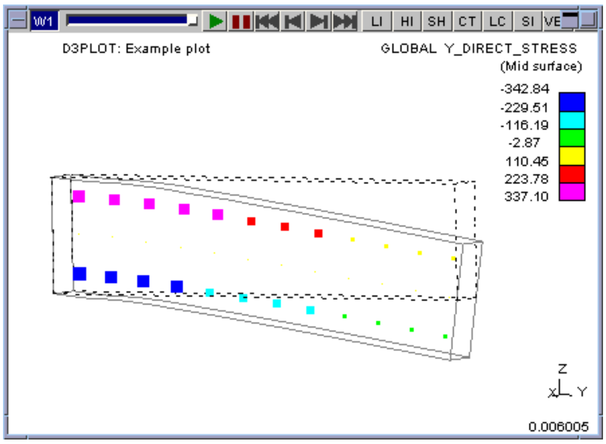
- Fixed, the default being 2 units
- Proportional to data magnitude, autoscaled to the current contour range.



Here is a simple cantilever of solid elements, loaded in bending, and plotted in CT mode.



And here is the same model, this time as a Cloud plot with variable symbol size.

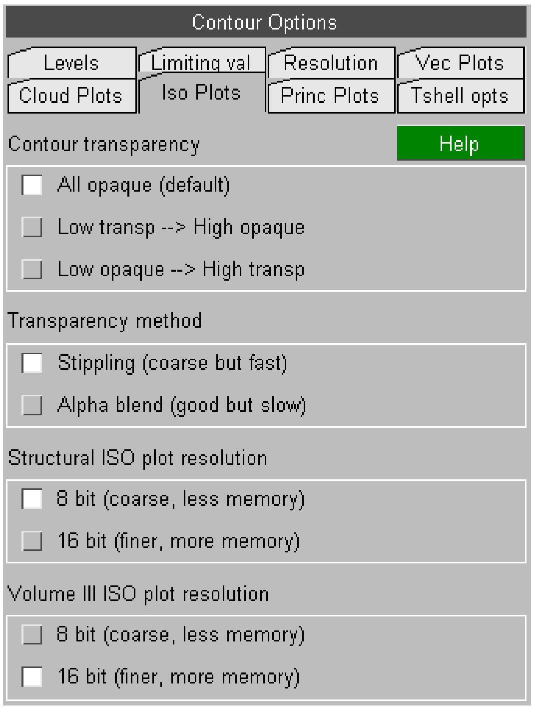


7.4.2.6. Iso Plots: Controlling the Attributes of Iso Mode Plots

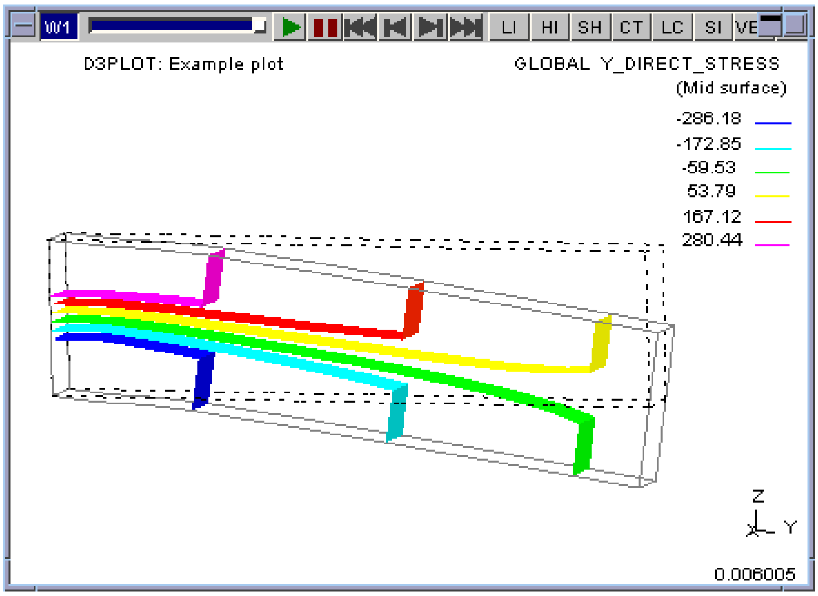
Iso Plots: Controlling the attributes of ISO mode plots

By default ISO plots are opaque, but this can obscure internal detail so it is possible to make them transparent.

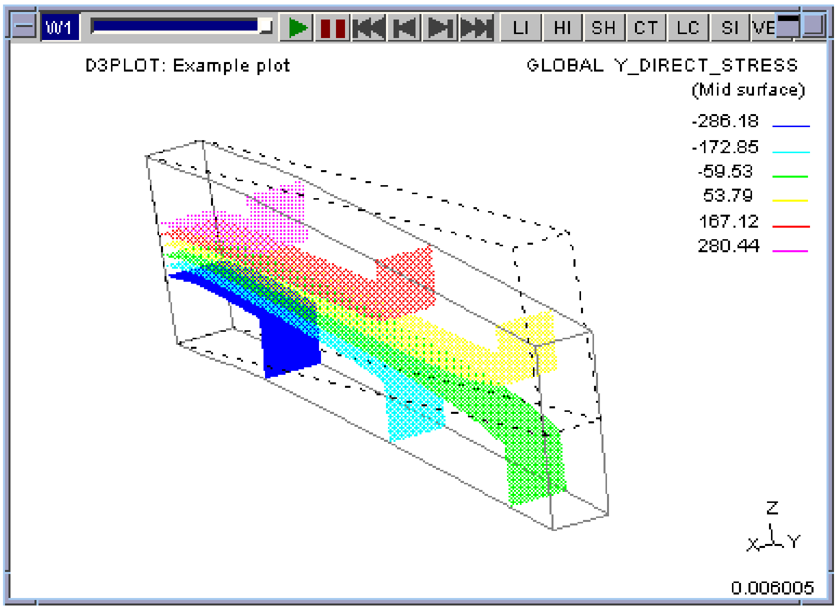
Transparency on graphics hardware is not perfect and, moreover, can be slow to render. So two alternative methods of displaying transparent ISO contours are provided.



Here is the same cantilever as above, rendered as a default (opaque) ISO plot:



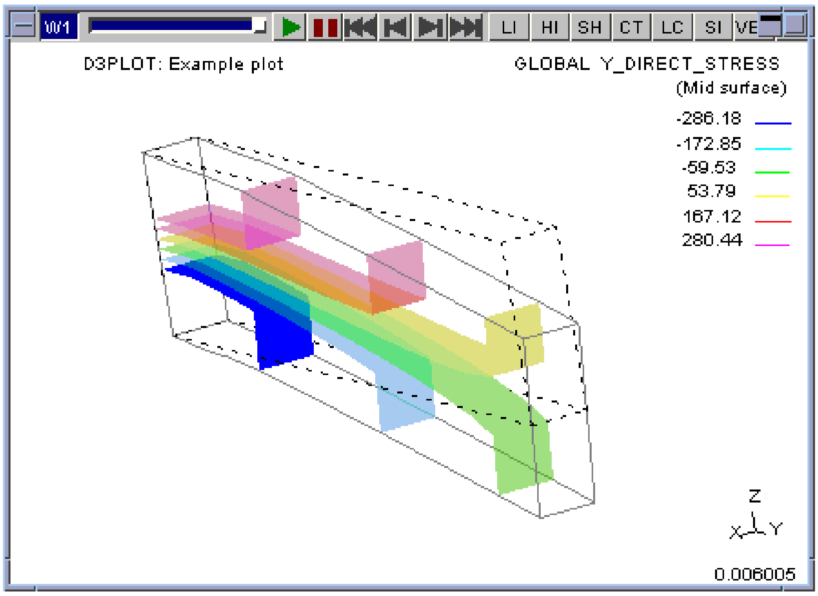
This is the same model using transparent "stippled" contours, with the lowest values (blue) rendered as opaque.



And here is the same image again, rendered using "Alpha blending".

The quality of the transparency is much better although it would be slower to draw with a large model.

However ...

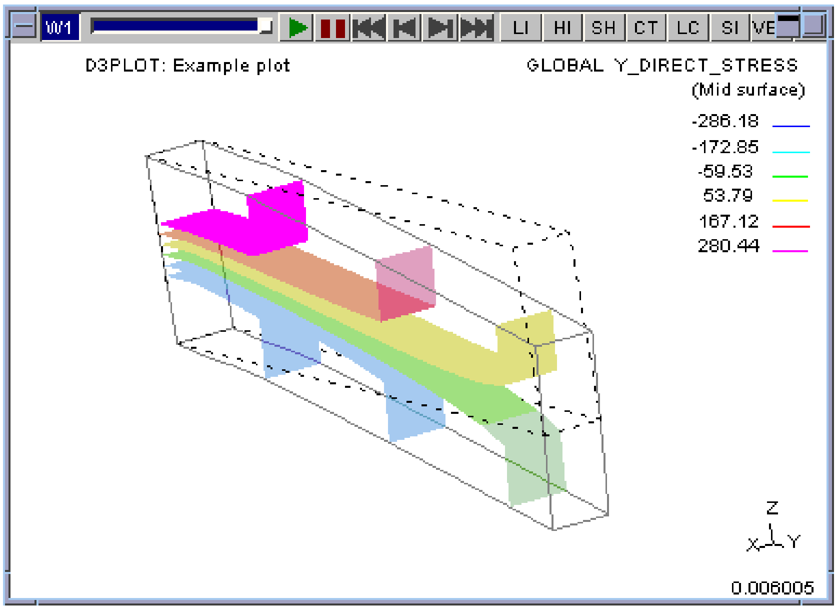


... using Alpha transparency does not always give the "right" answers.

Here is the same image, this time with the high values opaque, and it is clear that the lower colours are not visible when they are behind the higher ones.

This is because "Alpha blending" in the hardware is not, in itself, always a totally reliable way of producing realistic transparency because the results depend on the order in which facets are drawn.

It may be necessary to revert to stippling, which will always give "correct" precedence of facets - albeit with a poorer spatial resolution - to get acceptable results.



bit versus 16 bit Plot Resolution

8 bit gives a coarser result leading to a "bumpier" plot that is usually acceptable for structural data.

16 bit gives a smoother result that looks better, and is better suited to the sort of eulerian data used in Volume III type analyses. 16 bit plots use twice as much memory, which is why they are not the default for structural data plots.

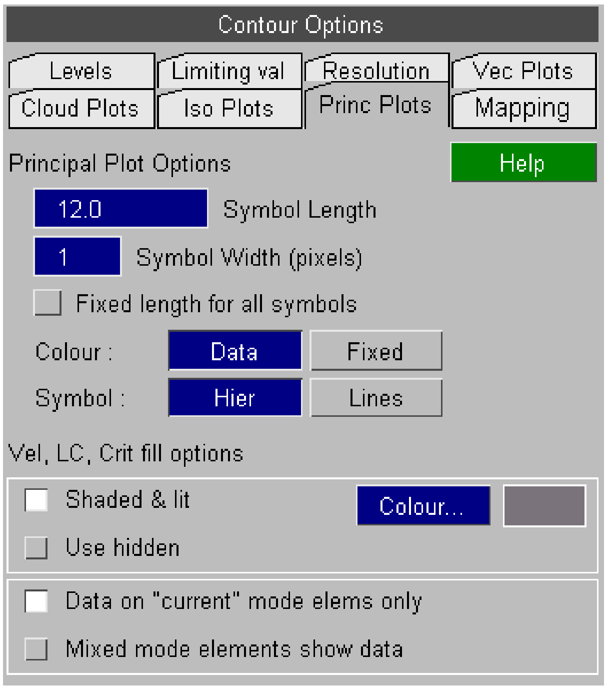
You can change these defaults via the preferences:

```
d3plot*struct_iso_resolution: 8 or 16
d3plot*vol3_iso_resolution: 8 or 16
```

7.4.2.7. Princ Plots: Controlling the Attributes of Principal Plots

Princ Plots: Controlling the attributes of Principal plots

This panel controls the plotting of principal stress and strain vector plots.



LENGTH/WIDTH : Setting vector length.

By default vector symbols are scaled by the data magnitude. **LENGTH** sets the maximum symbol length in model space units. To make it easier to see the vector symbols on some displays, the **WIDTH** of the lines used to draw the symbols can be increased. By default a line thickness of 1 pixel is used.

The **FIXED** option can be used to make all the vector symbols the same size regardless of the data magnitude. If this option is set then the colour of the vector symbols still represents the data magnitude.

COLOUR : Setting vector colour

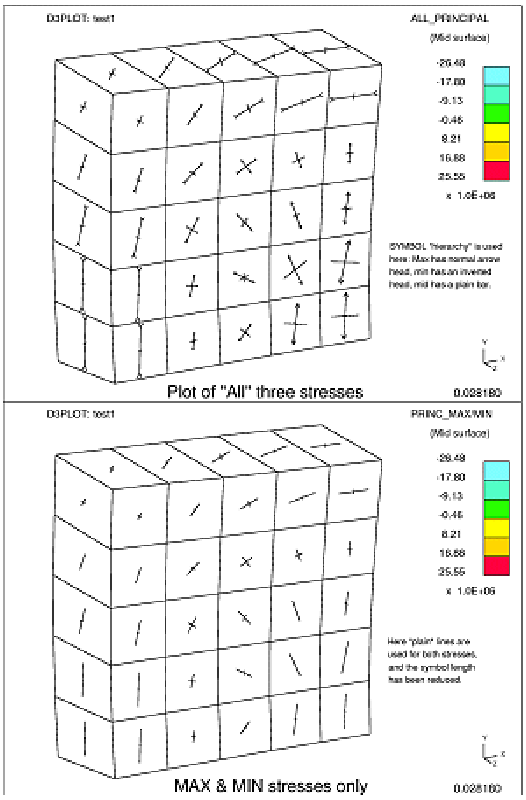
By default vectors are coloured using "contour" bands based on their data magnitude. This is the **DATA** option.

You can choose instead to use colour to distinguish between the components of multi-valued plots, the **FIXED** option.

SYMBOL : Vector symbol types

By default the vector symbols have a **HIER** archy: arrowhead for largest, inverted arrowhead for smallest, plain bar for middle.

You can choose to have plain **LINES** instead if you wish.



Vel, LC, Crit Fill Options

These options controls the appearance of element in the plotting modes where data vectors are imposed on top of "structure". For more information see [Vel, LC, Crit fill options]

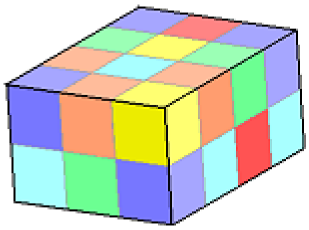
7.4.2.8. Mapping Data Onto a Cellular Grid for Plotting

Mapping data onto a cellular grid for plotting

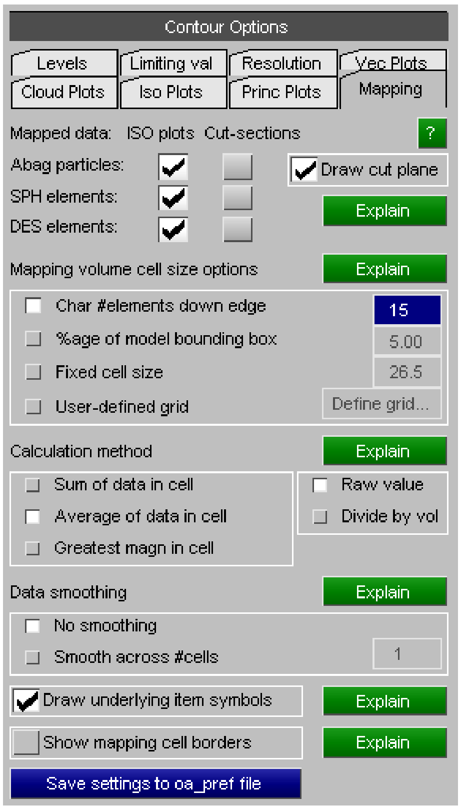
Some element types, notably Airbag particles but also SPH and DES elements, are "single point" and do not form a connected mesh, therefore contouring their results by averaging data across connected elements is not possible. D3PLOT performs contour plots by drawing each element in a single colour, but this is not always satisfactory since it can be hard to understand overall behaviour from a myriad of individual element results.

Data mapping attempts to solve this problem by imposing a cellular "sugar cube" mesh over the volume of space containing these elements, and computing values for each cell from the individual elements that occupy this cell. Since each cell in the mesh is connected to its neighbours it is possible to contour data across the mesh as a whole, and this can be especially useful for performing ISO surface plots, and also for plotting data on cut sections.

This image shows a notional distribution of "point" elements, coloured by contour value, distributed through a "sugar cube" cellular mesh.



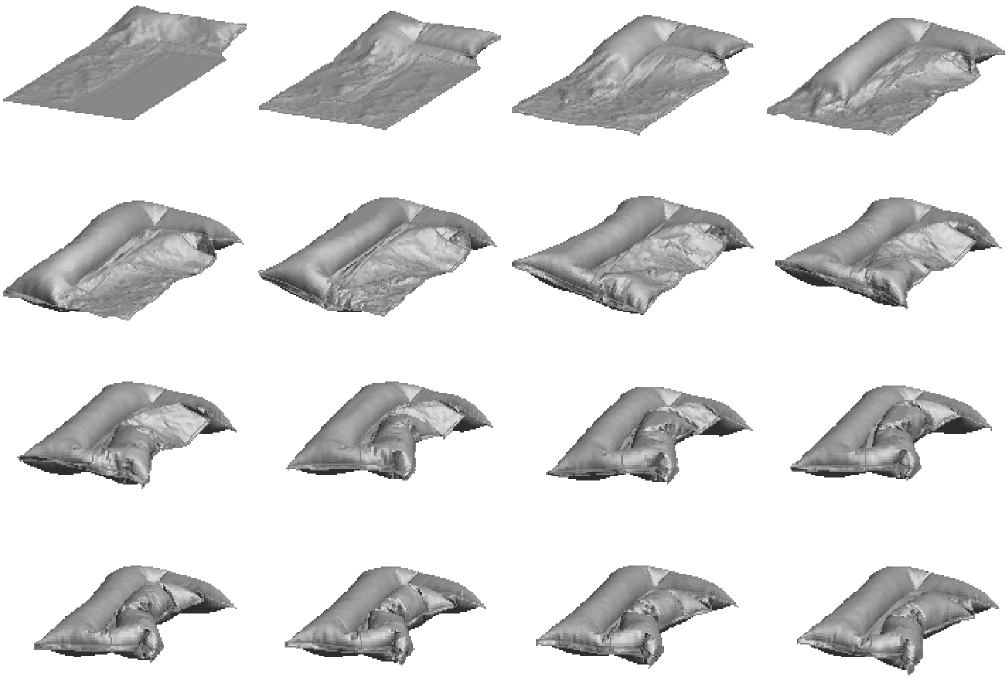
Performing some calculation to aggregate all elements in a cell into a single value might give a result like this.



The value of this process is best demonstrated by example. The following sequence of images shows a series of frames of an airbag being inflated using the CPM method (*AIRBAG_PARTICLE).

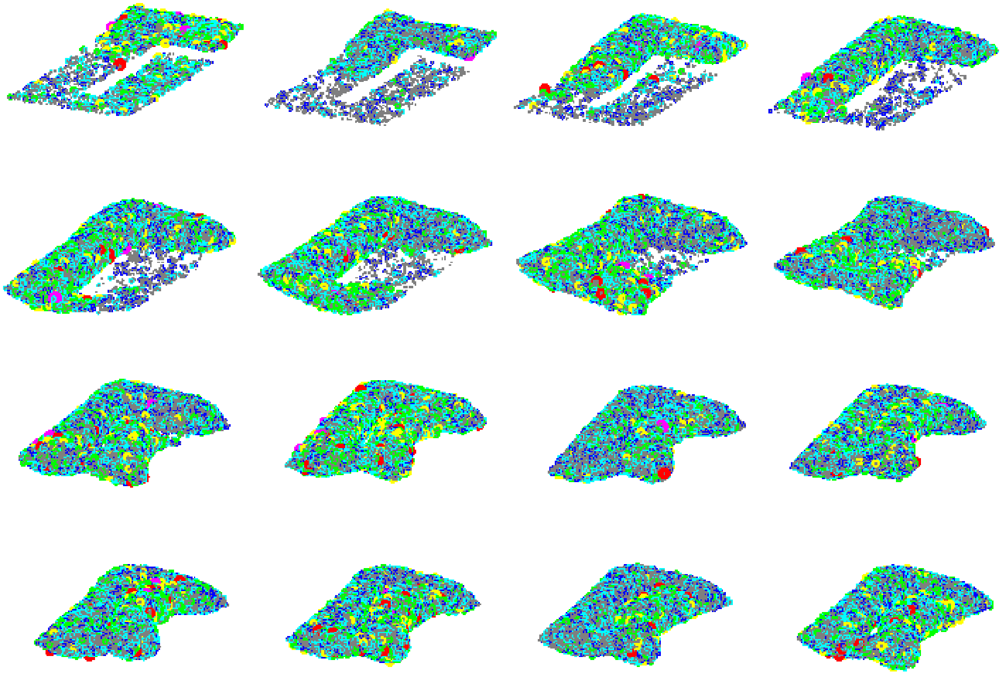
(1) Firstly a sequence showing the bag fabric being inflated.

The inflator is at the back, and the bag is a "U" shape in which gas has to travel towards the observer on the left hand side as viewed here, then across the near side, and finally away from the observer to fill the right hand side. The sequence below is approximately 16 milliseconds long with 1ms between frames, and just shows a shaded plot of the bag fabric..0000



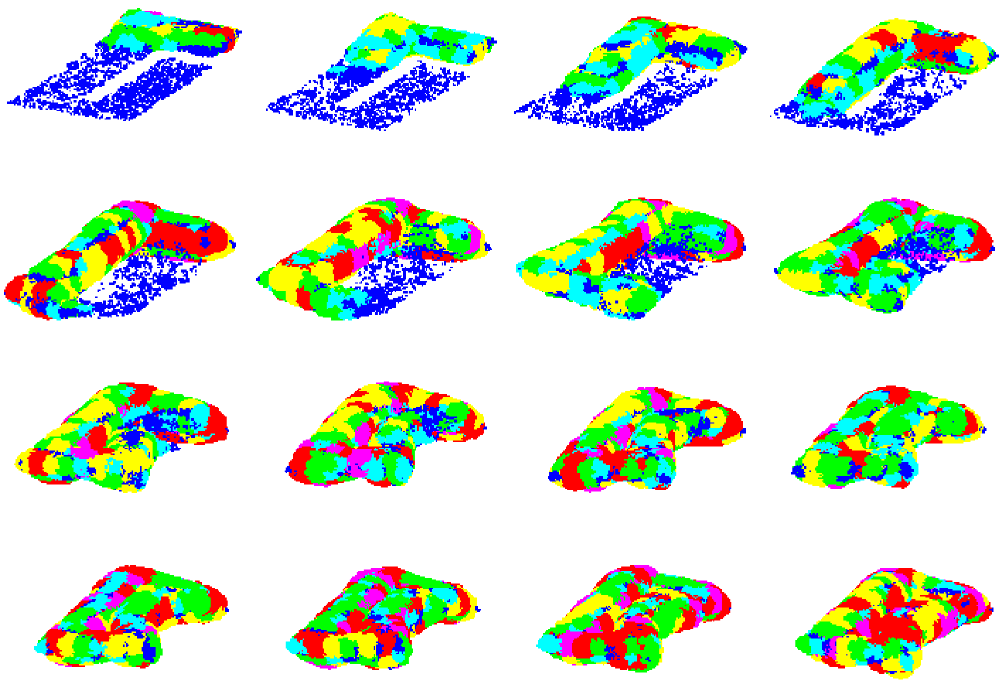
(2) Contour plot of particle translational velocity

Pressure in an airbag is a function of the sum of particle translational velocity within the volume "near" each particle, but it is clear from the sequence of images below that simply contouring particle velocity does not give a proper indication of pressure since each particle "bounces around" and it is impossible to get anything other than a very general feel for gas behaviour.



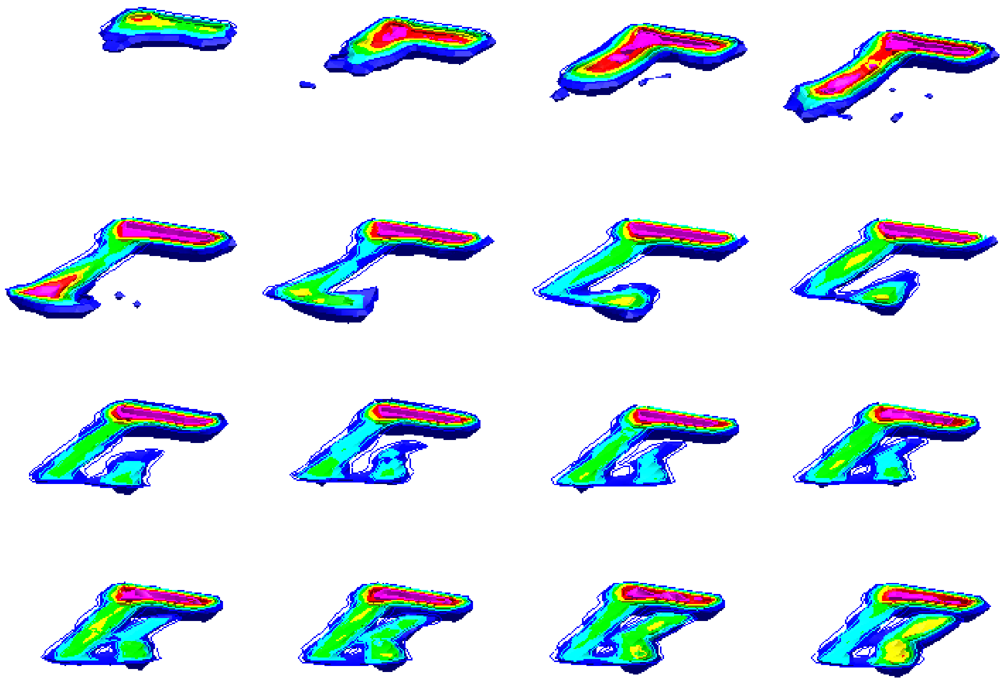
(3) Data-mapped display of Pressure calculated from the above

This plot shows the result of aggregating the particle velocities above into a cellular mesh, and calculating a pressure value for each cell. All particles in the cell are then given the same pressure value, which is why there are discrete blocks of colour.. A pattern is starting to emerge, but it is still hard to read.



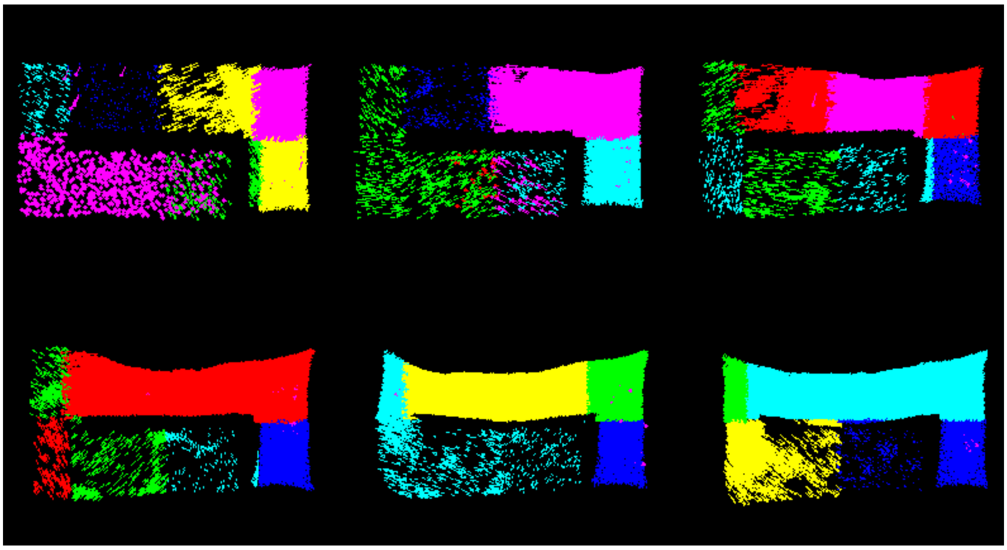
(4) ISO plot of PARTICLE_PRESSURE, with a cut section active

The plots below are the result of performing an ISO plot through the data in image (3) above, and cutting the bag roughly 1/2 way up its vertical axis to reveal the internal pressure distribution. It now becomes much easier to visualize the internal pressure and the flow of gas through the bag.

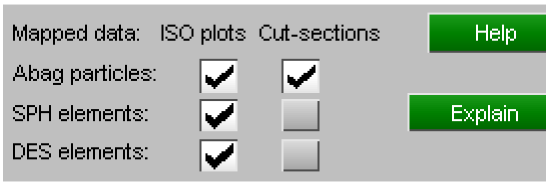


(5) Mapped velocity plot (plan view from above of the same bag, inflator on right)

In this case individual particle velocities have been mapped onto a cellular grid, then all particles within a cell have been assigned the average velocity vector in that cell. This gives an indication of gas flow direction and velocity.



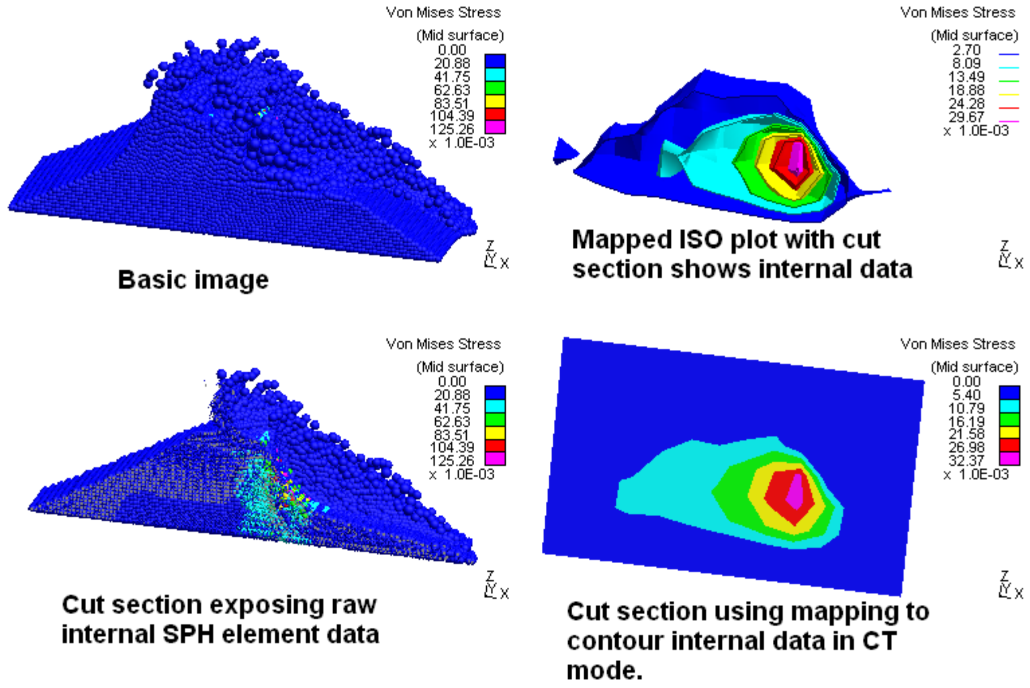
What data is actually mapped?



At present only "single point" element types can have their data mapped, which means airbag particles, SPH elements and Discrete Element Sphere (DES) elements.

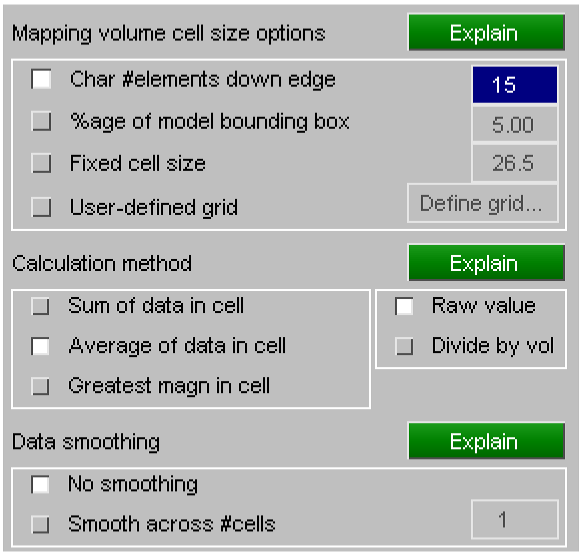
Experience has shown that ISO plots through all these element types work well, but that cut-sections through SPH and DES elements are normally best left unmapped, since while these two element types may not be connected in a mesh they are nevertheless usually modeled as a block of adjacent elements, and cut sections expose the inside of such blocks well.

Nevertheless it is possible to cut through these types and the image below shows the result of a sand dune modeled with SPH elements being hit by a vehicle. On the left display shows the raw SPH element data, and on the right data mapping has been used to create an ISO plot and a continuously contoured cut section through the sand.



How is mapping performed?

Different data component require different mapping methods, and the size of the cells used can also influence the outcome. Three settings control mapping:



(1) Cell size.

When data is mapped D3PLOT will automatically calculate the bounding box that encloses the relevant elements, and this box is then sub-divided into cubical cells. The size of these cells can be controlled in the following ways:

Fixed cell size	The user sets an explicit cell size in model space units.	enclose the bounding box around the elements being plotted.
User-defined grid	The user sets the origin coordinate, cell size and #cells in each of the [X,Y,Z] dimensions.	In this mode the grid remains fixed in model space over all results states.

It may be necessary to experiment to experiment a bit to find the best cell size for a given model, and different data components may also benefit from different cell sizes.

(2) Calculation method

Sum of data in cell	Simple sum of the data values of all the elements that lie in a cell
Average of data in cell	The sum as above, but divided by the number of elements in the cell.
Greatest magnitude in cell	The single value with the greatest magnitude of all elements in the cell.

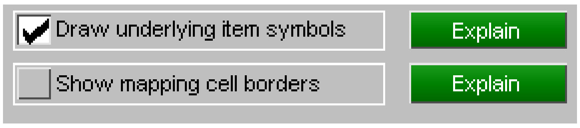
It is also possible to decide whether to use the raw value as calculated by the method above, or to divide this value by cell volume.

Note : certain "built in" mapped components (airbag particle pressure and velocity) will temporarily override these settings in order to calculate the correct values, and may have further internal factors as required to obtain the correct results. See Other values calculated by D3PLOT for more information.

(3) Data smoothing

The mapping process itself introduces a degree of smoothing into the results, but further smoothing can help with some data components. By default no smoothing is performed, but if turned on data in each cell will be a weighted average of the data in the cell itself plus that in the N cells that surround it.

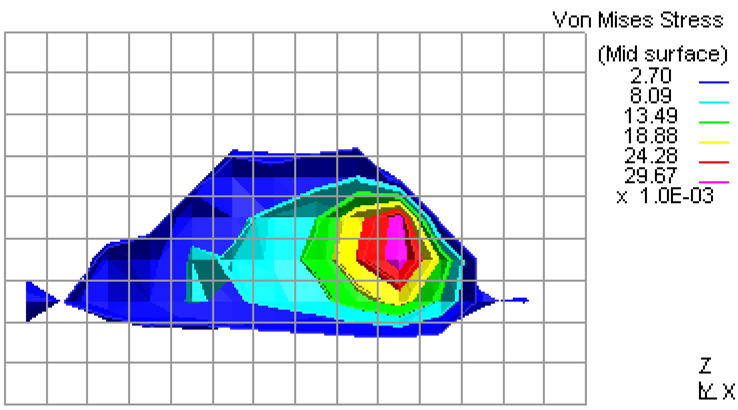
Further data mapping options



Draw cut plane	When a cut-plane is active, and this data type is set to be displayed on it (see above) the default is for the cut plane itself to show contoured results in the appropriate plotting mode. However in some cases, especially thick cuts through airbag particle plots, it can be helpful to omit the cut plane itself so that the "sliced data inside the bag" can be seen more clearly, and this option allows display of the plane to be switched off.
Draw underlying item symbols	By default the actual element symbols, for example airbag particles, will be included in data mapped contour plots. Sometimes this is a nuisance, particularly in ISO plot, so deselecting this option will suppress these symbols in those plots only.
Show mapping cell borders	Normally the cells used for mapping are invisible, but it can be useful to visualise them when trying to work out what is happening. Selecting this option will display the matrix of cells. An example is shown below.

This shows the section through the SPH mesh used in the example above, but this time with cell borders visible.

This is an orthogonal XZ view, so only a 2D grid is seen, an isometric view would should the full 3d grid.



7.4.3. MAX & MIN Displaying Max and Min Values on Plots

MAX & MIN Displaying max and min values on plots

D3PLOT can calculate the maximum and minimum <n> values on the current plot, and display it at the top of the graphics window. Element-derived data components show element values, and nodally-derived ones show nodes. Element values are taken either from the element centre values or the element integration point values (for models with results at multiple integration points). Consequently the max/min values can differ slightly from the max/min values on the contour bar if the contouring resolution is not unaveraged.

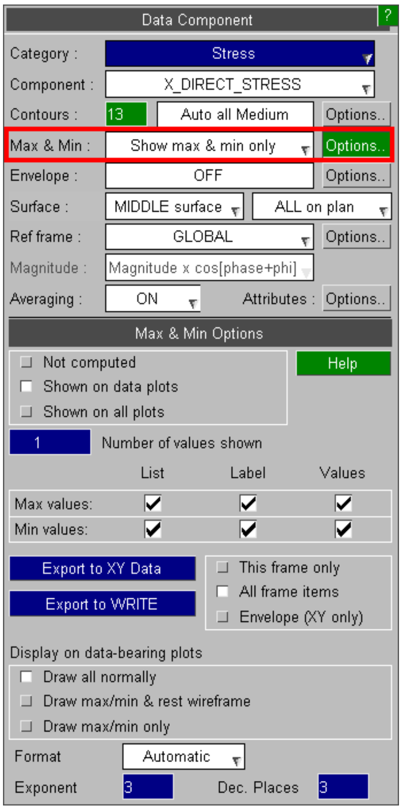
If a model contains 3D elements (Solids and Thick Shells) then by default D3PLOT only uses the values on external visible faces for the contour bar and ignores internal faces for SI, CT and LC data plots (This behavior can be changed - see Visible faces vs All faces).

When displaying the Min and Max values on a plot D3PLOT will report both visible maximum and minimum values and if an internal face has a higher or lower value it will also report the element that value belongs to in the top left of the screen.

Not computed	No max/min values computed or displayed
Shown on data plots	Max/min values are computed, but only displayed on data bearing plots (CT, SI, etc)
Shown on all plots	Max/min values are shown on all plot types. For non data-bearing ones the values shown are those of the currently selected data component.

Number of values shown	By default only 1 of each max and min is shown. You can choose any number, but space on the plot is limited and a practical limit is about 30 pairs of values. Very large numbers will also take longer to compute and store.

List/label options	By default both values are listed on the plot, and the relevant elements/nodes are labelled to identify them along with the values. All are switchable.



Export to XY Data

This option will calculate the max and min values for all states in the analysis, and export them as (x,y) data to the XY_DATA tool for graphical plotting. The actual results sent depend on the settings below:

This frame only.	Generates curves only for the <n> items that are the max and min in this frame. This will result in <2n> curves.
All frame items.	Generates curves for the <n> items that are the max/min in each state. This can result in up to 2 x #states x <n> curves if the max/min items in each state are different.
Envelope	This produces only two curves, a maximum and an minimum. The X axis is state time, and the Y axis is the actual max/min value at each state regardless of the actual element/node it comes from.

Export to WRITE

This option is similar to **Export to XY Data** , except that it builds the list of elements/nodes based on the options selected, and exports them to the **WRITE** tool for numerical output at the current time.

Display on Data-bearing plots:

Draw all normally	Normal plotting with no restrictions
Draw max/min and rest wireframe	The max and min <n> items are drawn in the current plotting mode, and the rest in wireframe mode
Draw max/min only	Only the <n> max/min items are drawn in the current plotting mode.

7.4.4. ENVELOPE...

ENVELOPE...

Envelope plotting can be used to plot either the minimum, maximum or the absolute maximum data values within a range of states.

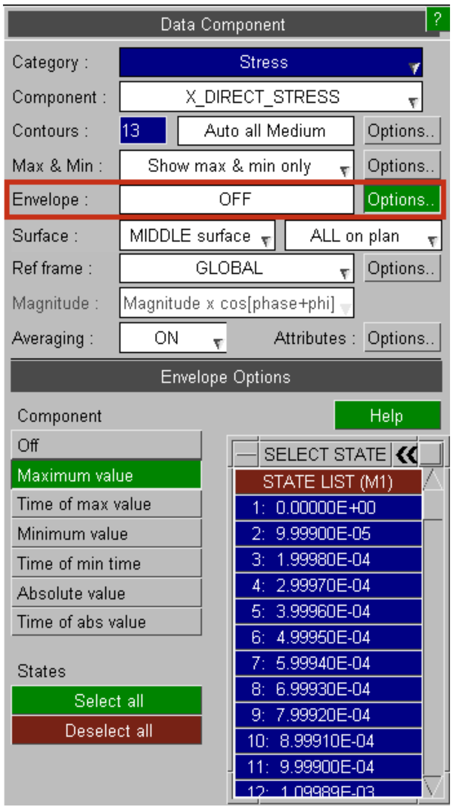
In addition to plotting the minimum or maximum values the times of the states that the values occurred at can also be plotted.

If envelope plotting is turned on for an element based data component (i.e. strain) then element averaging is automatically turned off (see AVERAGING... Controlling data averaging across adjacent elements).

When envelope plotting is turned on Line Contours (see LC) are only available for node based data components (ire velocity).

Envelope plotting may also be used with the **WRITE** option (see WRITE Listing Numerical Data to Screen and/or File) but it is not available during **ANIMATION** (see Animation How to Display, Control, Store and Retrieve Animation Sequences) or when the **REFERENCE_STATE** option (see REFERENCE STATE/MODEL) is being used.

Note: At present **ENVELOPE** plotting only functions in 2D mode. Users running in 3D under OpenGL will be temporarily swapped back to 2D mode for the duration of an envelope plotting operation.



7.4.5. SURFACE / INT Point

SURFACE / INT Point

This option can be used to select which surface data is plotted for for Shell and Thick Elements and how multiple integration point data is plotted for Shell, Thick Shell and Solid Elements. If the model contains fully integrated shells with results output at each on plan integration point (MAXINT set to a negative value on the *DATABASE_EXTENT_BINARY card) or fully integrated solids with results output at each integration point (NINTSLD set to 8 on the *DATABASE_EXTENT_BINARY card) then an option can be set to plot all the individual on plan points / solid integration points (default) or average them to a single point per shell / per solid and then plot. By plotting all integration points, the variation of data across the Shell or Solid can be seen.

If the current data component is a Beam Component then this option can be used to select the Beam Integration point.

Shell Surfaces

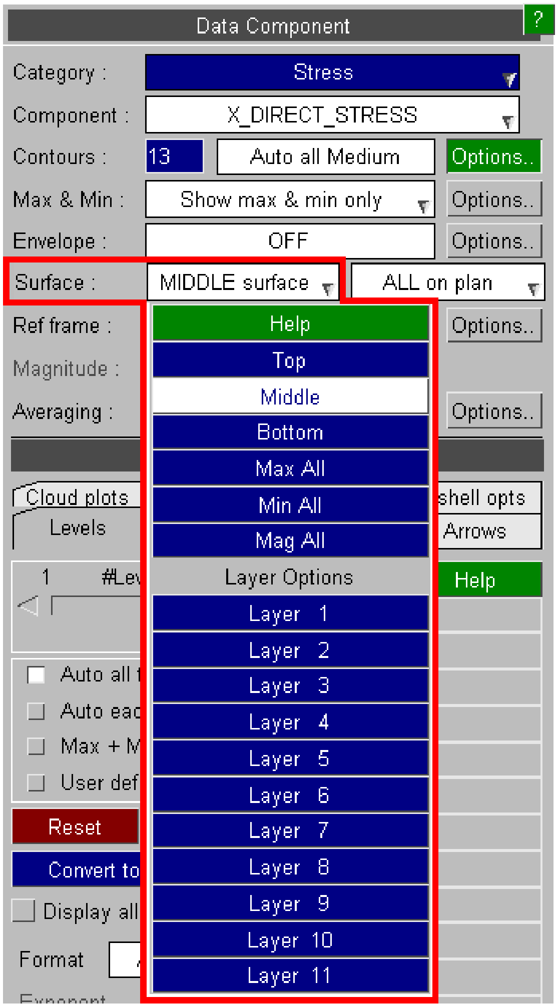
This figure shows the shell integration point selection box, which applies to stress and strain tensor derived results.

By default shells write results at 3 "surfaces":

Top	Is the outermost (most +ve local Z) integration point.	Note that when using the default Gaussian integration rules in LS-DYNA the inner and outer integration points of shells are ***NOT*** the "outer fibres" of the element. For a fuller explanation see the WARNING below.
Middle	Is the neutral axis, i.e. mid-plane.	
Bottom	Is the innermost (most -ve local Z) integration point.	

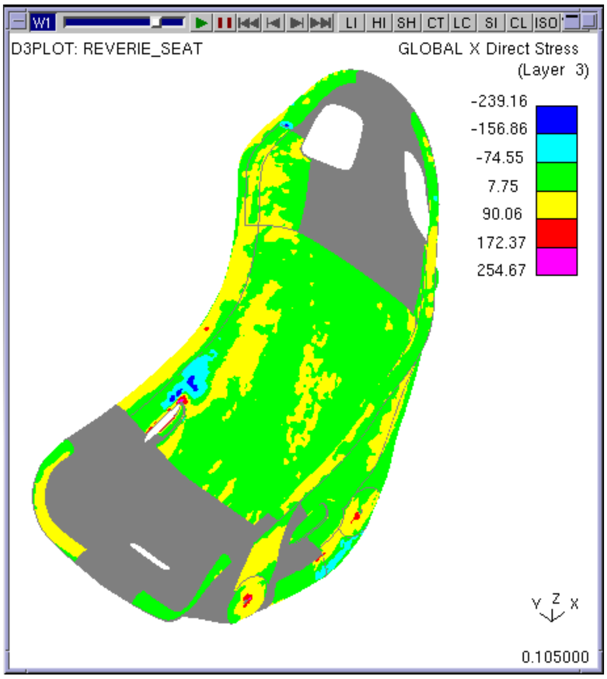
In this example shell output has been defined with < **`maxint`** > = 11, giving the option of each of the 11 integration points in the element.

Before D3PLOT 11.0, the layers were in the order of the integration points output by LS-DYNA, e.g. for < **`maxint`** >=3 Layer 1 was the MIDDLE surface, Layer 2 was the BOTTOM surface and Layer 3 was the TOP surface (see Thin shell integration points). From v11.0 onwards Layer 1->Layer n is always Bottom->Top (so long as a `.ztf` file is present).



From Version 13 onwards, if there is no data for the selected integration point the shells are greyed out.

In this example "Layer 3" is selected, but some shells only have data at two integration points.



Normally you will be interested in results at a given integration point, but it is also possible to extract the following values scanned from all integration points through the thickness of the element and the derived mid-surface value if the element does not have an integration point at the mid-surface:

MAX_ALL	Finds the maximum (most +ve) value
MIN_ALL	Finds the minimum (most -ve) value
MAG_ALL	Finds the +ve or -ve value with the greatest magnitude. Result may be +ve or -ve since the calculation is: `if ( |val| > |curr| ) curr = val`

It should be noted that when shell results for surfaces MAX_ALL, MIN_ALL or MAG_ALL are calculated at nodes, first the nodal values (averaged over the connected elements) at each integration point are calculated and then these values are scanned to find the max, min or mag value. This means that the max/min/mag value at a node can be different to the average of the corresponding max/min/mag values of the connected shells as the shell values may not all come from the same through thickness integration point.

Shell integration point data written from LS-DYNA

There are two issues to be considered here:

1. The number of through thickness integration points in the shell element formulation.
2. How many integration points worth of data are written to the database files.

Unfortunately these two parameters are not directly related in LS-DYNA, and have to be set independently:

- Shell element through-thickness integration points are controlled on the *SECTION_SHELL card. (Composites are different)
- The number of "surfaces" output is controlled by parameter < `maxint` > on the *DATABASE_EXTENT_BINARY card

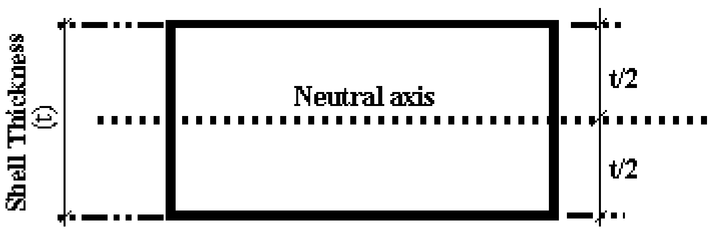
By default (< `maxint` > = 3) data at three "surfaces" (top, middle and bottom) are written for all shell elements, regardless of the actual number of integration points in any element formulation. Normally this is satisfactory, since in most models with significant plastic strain the detailed distribution of stress and strain through element thickness is not that important, but where models remain mainly elastic or where composites are used it may be necessary to set a different value. See Thin shell integration points for a fuller discussion of these parameters and how they affect output.

WARNING:

In LS-DYNA analyses the top and bottom shell "surfaces" are **not** the outer fibres of the element if the default Gaussian integration scheme is used: they are located some way in from the outer fibres.

The following table shows the location of the outermost integration points, as a function of shell half thickness (`t/2`), for the most commonly used numbers of points.

No of Points	Distance from neutral axis as a proportion of `t/2` (Gaussian integration)
1	0.0 (membrane)
2	0.577
3	0.775
4	0.861
5	0.906
6	0.932



Where you have written an odd number of integration points to the output file the "mid surface" will be the mid point.

Where you have written an even number of points it will be the average if the two "middle" values. For example if you write 6 points it will be averaged from #3 and #4.

Note 1: While thick shells write results at surfaces too, by default this flag has no effect when they are plotted since their (visual) thickness permits all three surface results to be displayed simultaneously on their respective faces. This can be changed so that each surface is plotted separately, as described in Contour Display Options for Thick Shells

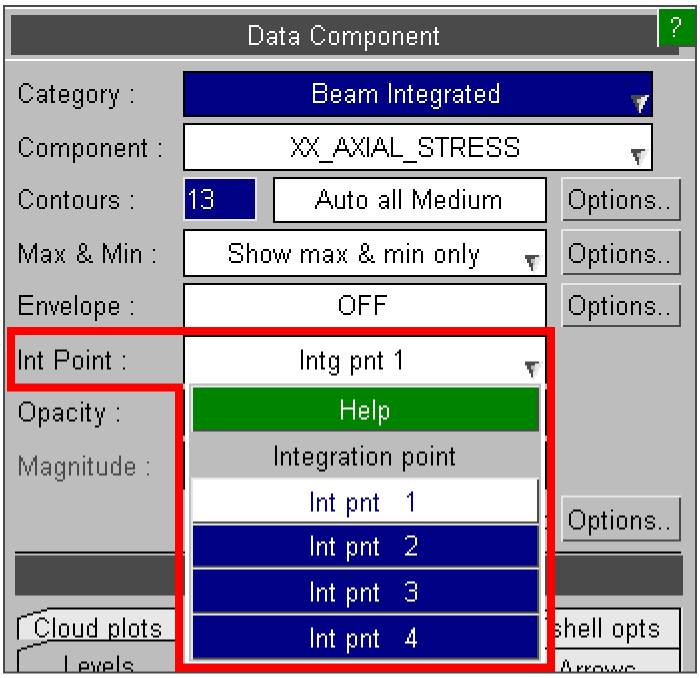
Note 2: Historically LS-DYNA has reverted to trapezoidal integration for 6+ integration points, although this is undocumented and the author has a sneaking suspicion that more recent versions of LS-DYNA may use Gaussian integration for up to 11 points. Examine such results with care!

A more detailed description of shell output, with particular reference to "surfaces", "layers" and integration schemes, is given in Thin Shell Integration Points .

Beam Integration Points

This figure shows the beam integration point selection box, which applies to results from "integrated" (Hughes-Liu etc) beams.

The "extra" **INTEGRATED** results for these beam types only are written at the specified integration points. Only one can be plotted at a time, and this is selected here.



7.4.6. SURFACE With Composite Plys

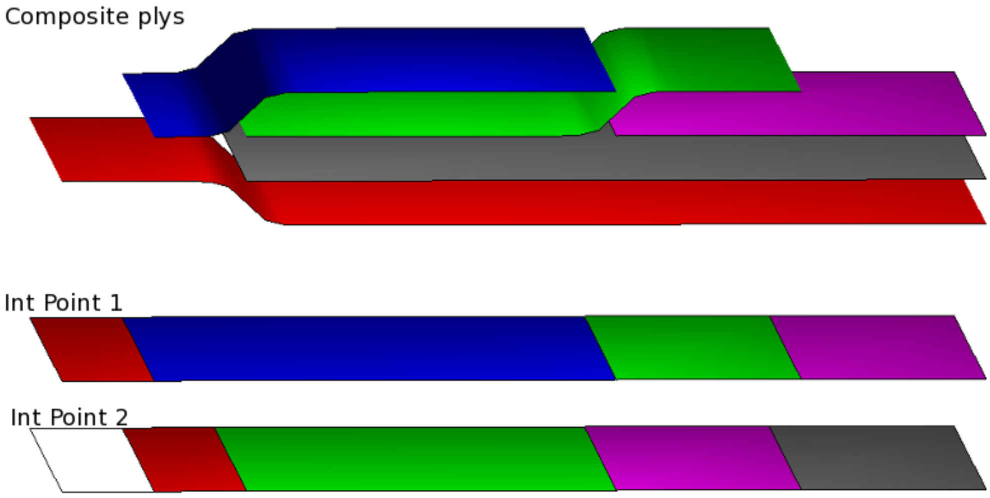
SURFACE with Composite Plys

From D3PLOT 13.0 onwards, where composite plys are available, data can be plotted on a surface composed of plys.

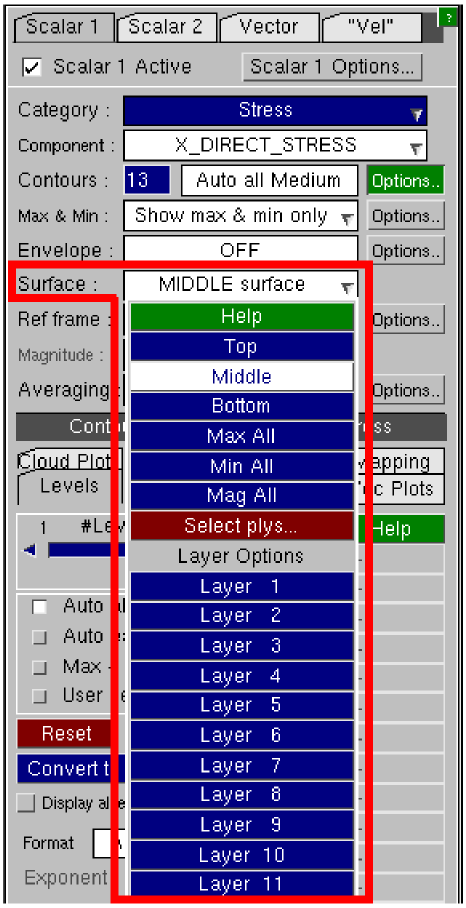
This requires composites to be set-up in PRIMER using the Composites tool, or equivalently *SHELL_COMPOSITE_LONG cards, and a `.ztf` file. (Composite plys created using a *PART_COMPOSITE card are not available using this feature.)

The surface is only applicable to Shell elements with stress tensor derived results, plastic strain, and extra results.

For Shells with composite plys, integration point data may not be appropriate. For example, the sketch opposite shows a strip of material containing five different plys. If data is extracted by integration point, the data may be across blocks from different plys with different material properties.



If composite plys are available, in the shell integration point selection box there is a further option to **Select plys...** which displays the Ply Selection menu.

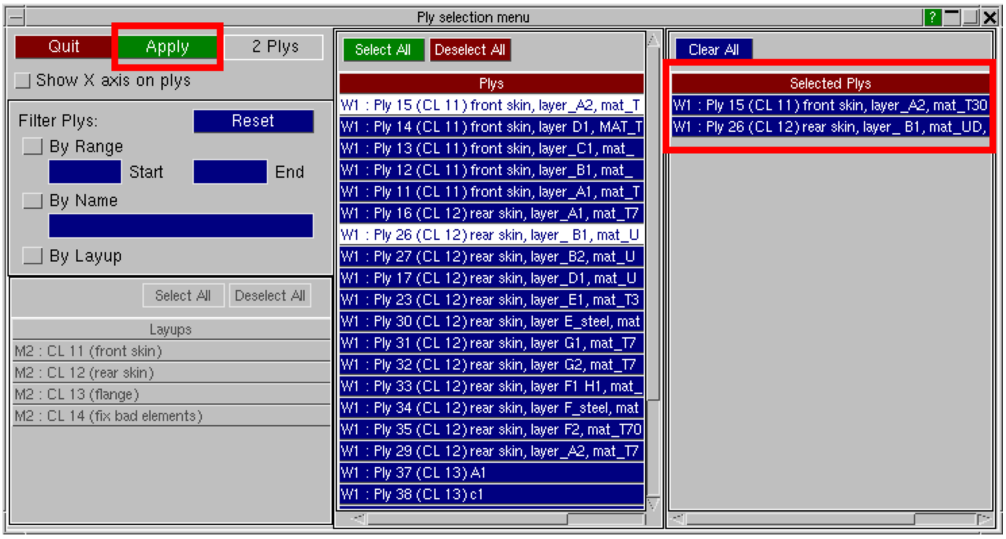


Note 1:	If Shell elements are in more than one selected ply then the ply with the lowest ID is used.
Note 2:	Data will not be averaged across different plys.
Note 3:	Elements which are not contained in the selected plys are greyed out.

In the Ply Selection menu the selected plys are listed on the right hand side.

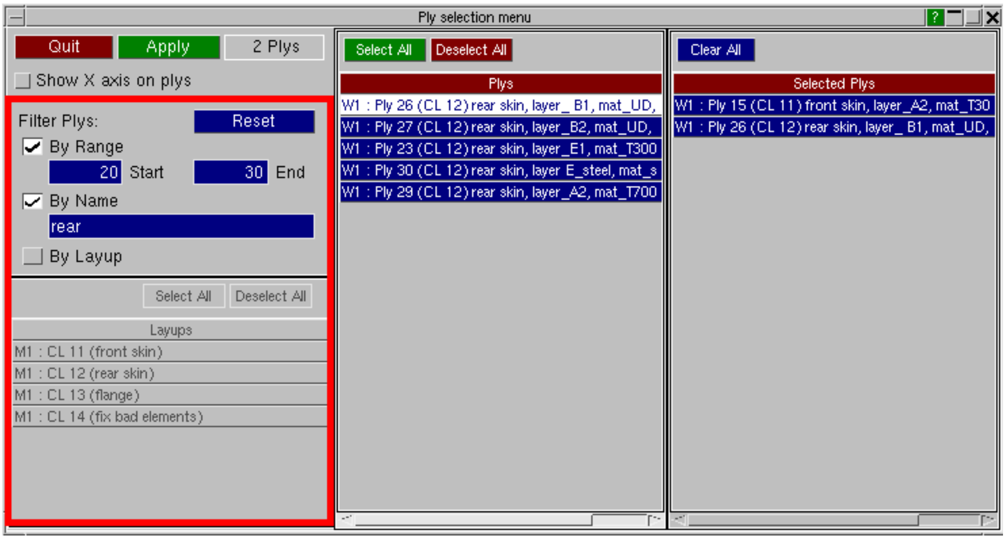
To add and remove selected plys click on the list of available plys.

To confirm the selection the user must press **Apply** .



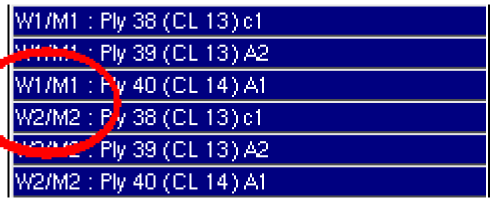
The list of available plys can be filtered using the options on the left:

By Range	Ply IDs must lie within the given range.
By Name	Ply names must contain this text (case insensitive).
By Layup	Plys must be contained in selected layups. (Only available if layups have been set-up in PRIMER.)
Reset	Deselects **By Range** , **By Name** and **By Layup**, and clears start, end and name fields, and any selected layups.



The menu includes plys in all active windows.

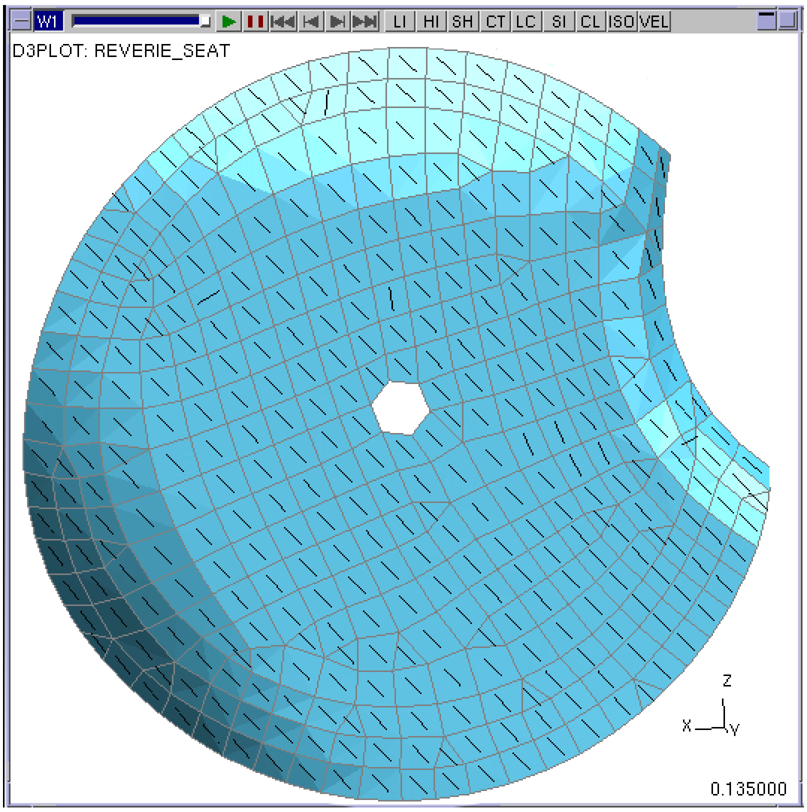
The plys are ordered by window ID, model ID, then layup ID and position in layup (where layups have been set-up), or by ply ID (if layups are not available).



Ply local X axis

Once the user has selected plys, **Show X axis on plys** can be used to see the ply local X axis on the selected plys.

Note, because Shells can be in multiple plys, D3PLOT requires that the user selects plys to see the ply local X axis.



Local X axis on Selected Plys

7.4.7. REF_FRAME... Choosing the Frame of Reference

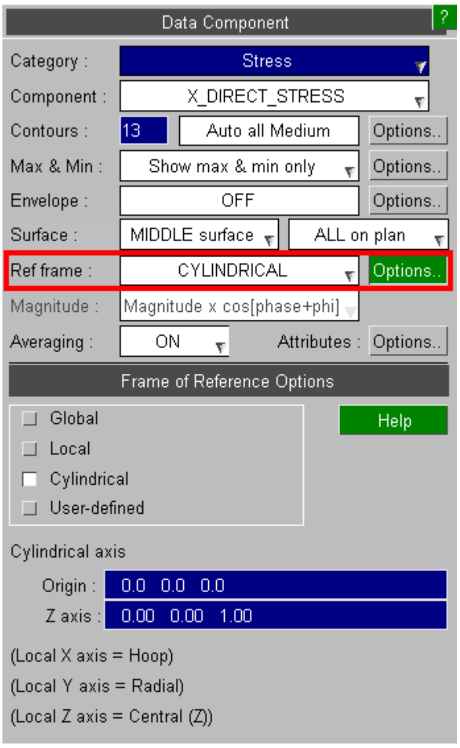
REF_FRAME... Choosing the frame of reference

Directional data components may be plotted in model **Global,** element **Local** , (global) **Cylindrical** or **User-defined** coordinate systems. The default is **Global** .

If the model has composite plys there is also a **Ply Local** option. This rotates the element local so the X' axis is given by the ply beta angle. The **Ply Local** frame of reference is only applicable if the Surface is **Selected Plys** (see Sections [SURFACE with composite plys](#) and [Ply local X axis](#)). The option **Show X axis on plys** sketches the ply local X axis on the Selected Plys.

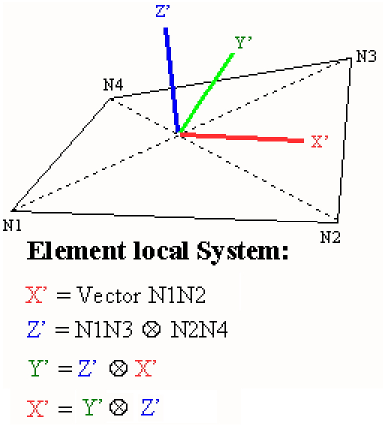
Directional components are basic stresses and strains from their respective tensors, i.e. **`X_DIRECT_STRESS`** , **`SX_DIRECT_STRAIN`** etc. See Section 12 for more details.

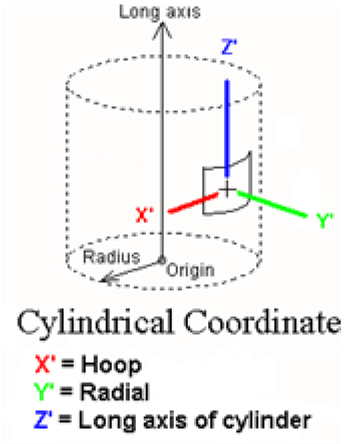
In this example a **Cylindrical** system has been used, and the user has defined the origin and vector of the local Z' axis.



The element local system is computed from its topology. A four noded element is shown here, for a 3 noded element Z' is normal to the (flat) plane N1N2N3.

A cylindrical system is only suitable for elements that do genuinely lie in the plane of a cylindrical wall. The Y' vector is perpendicular to the Z' axis through the element center.





Element local axes are calculated as follows:

[**X'**] (approx)	From vector **N1N2**
[**Z'**] (outward normal)	From cross product **N1N3** x **N2N4**
[**Y'**]	From cross product [**Z'**] x [**X'**]
X'] (warping correction)	From cross product [**Y'**] x [**Z'**]

Let [**U**] be the vector from the cylinder origin to the element centre, then axes are calculated as follows:

[**X'**] (hoop)	from cross product [**U**] x [**Long axis**]
[**Y'**] (radial)	from cross product [**Long axis**] x [**X'**]
[**Z'**]	is the same as the [**Long axis**]

The data to which coordinate system transformations are applied

- **Element tensor derived data** is transformed to Element Local or Cylindrical.

  "Tensor derived" means Stress and Strain tensors, and any User-defined tensor components.

- **Nodal vector derived data** is transformed to Cylindrical only. ("Local" has no meaning for nodes)

  "Vector derived" means Displacement, Velocity and Accelerations vectors, and any User-defined vector components.

One exception is that vector plots of nodal vector data are always presented in the global system in order to show the "true" vector directions.

7.4.8. MAGNITUDE

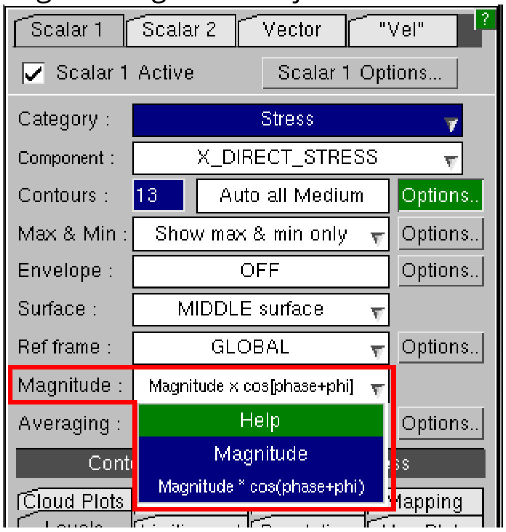
MAGNITUDE

For frequency domain models, where results depend on phase angles, by default D3PLOT scales results based on the current value of phi using the formula:

*val = magnitude * cos(phase + phi)* .

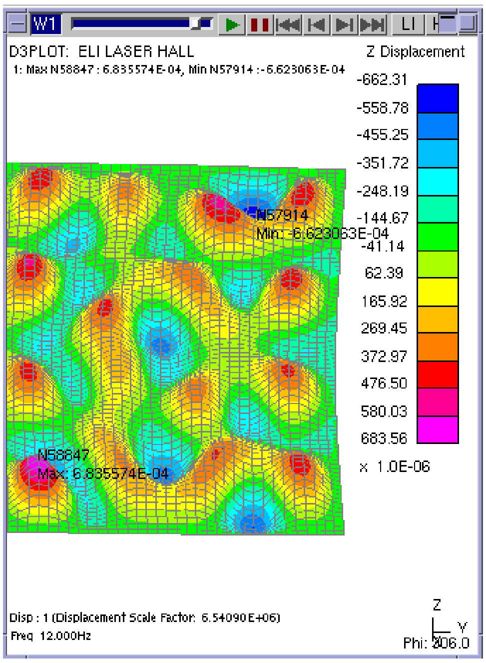
This means that it is not possible to see the maximum values on all elements and nodes at the same time, unless they are in phase.

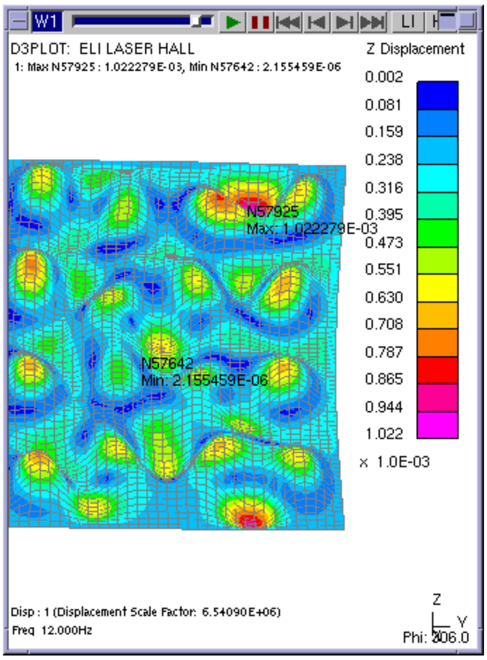
However, it possible to change to *magnitude* to just see the maximum values instead.



Z displacement at phi=306.0 scaled by *magnitude*cos(phase + phi)* :

Z displacement at phi=306.0 scaled by *magnitude* :



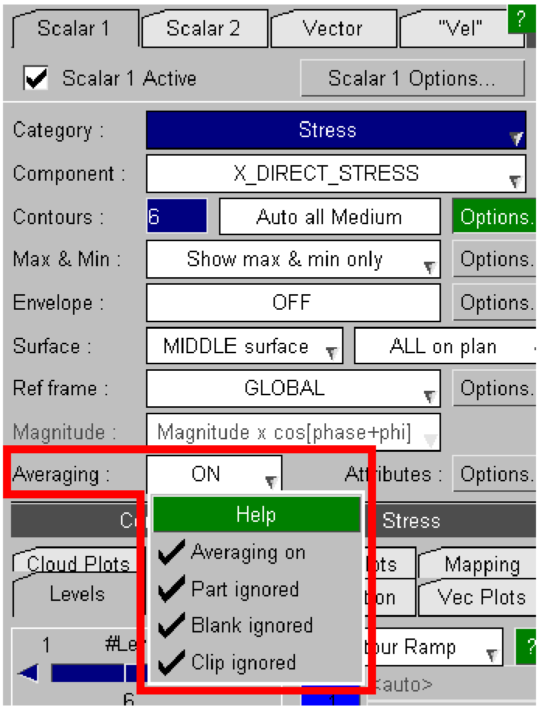


7.4.9. AVERAGING... Controlling Data Averaging Across Adjacent Elements

AVERAGING... Controlling data averaging across adjacent elements

By default data for contour plots is averaged across adjacent elements of the same type, regardless of their material, etc.

These settings allow you to modify this behaviour.



The averaging control settings have the following meanings:

Averaging Switches averaging on or off (default on). If turned off then no data averaging occurs for contour plots: this results in a "patchwork quilt" effect for area contour plots and no results at all for line contour plots.

Part ignored By default this is on and the part is not considered when averaging. If you switch this off then averaging will **not** take place across adjacent elements if they are in different parts.

Blanking ignored By default this is on and blanked elements are still included in the averaging process even though they are not visible. This means that blanking elements will not change the values used for contouring on those that remain. If you switch this off then results from blanked elements are **not** included when results are averaged at nodes.

Clipping ignored By default this is on and elements that have been volume-clipped from the display are still included in the averaging process even though they are not visible. If you switch this off then results from volume-clipped elements are **not** included when results are averaged at nodes (i.e. same logic as is applied to blanked elements above).

Notes on data averaging:

Note 1: Averaging of element data at nodes for contouring only takes place for element derived data, e.g. stresses. Where the data being plotted

	is nodally derived, e.g. velocity, then averaging is not used and the settings above have no effect.
Note 2:	Averaging has an effect beyond plotting: it can also influence how element-derived scalar data is computed at nodes for **WRITE** and **XY_DATA** output.
Note 3:	Averaging never takes place over dissimilar element types. For example where a node is common to both a solid and a shell data at the node is computed separately for the "parent" element types, even if the data component type is valid for both types.
Note 4:	Averaging is applied if requested, even if it might not be sensible to do so: this can be an issue when directional components are plotted in element local systems. For example if `LOCAL X_DIRECT_STRESS` is used where two shells meet at a right angle (i.e. a flange meets a web) you may be averaging stresses in directions that are 90 degrees apart.
Note 5:	A related error is to average across shells, using top or bottom surface data, when adjacent shells have inverted surfaces; i.e. their outward normals (local Z axes) point in opposite directions. This is usually the result of a meshing error, and it can produce strange contours. (To check outward normals turn on the element local triads with `DISPLAY_OPTIONS` , `LOCAL_TRIAD;` or do a continuous-tone (**CT**) plot of the element outward normals using (geometric) component `ON_OUTWARD_NORMAL` .)

7.4.10. OPACITY_SWITCH... Making Overlying Structure Transparent

OPACITY_SWITCH Making overlying structure transparent

This option is only displayed if the currently selected data component is one of the following:

Beam
Spring
Spotweld
SPC
Seatbelt
X-Section
Load Path
ICFD
CESE

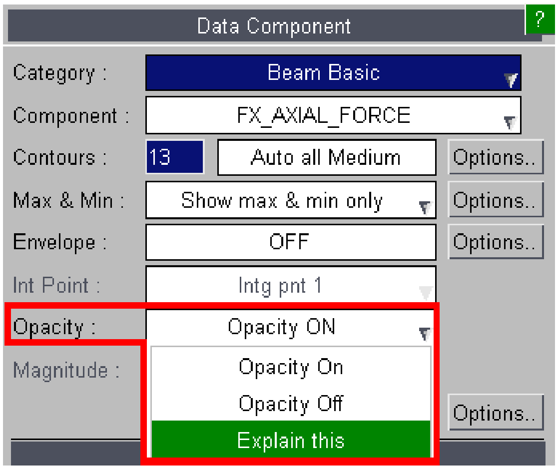
All of these entity types are often buried inside a model which also contains shells and solids, and it can be difficult to see them because of the intervening structure. Therefore D3PLOT allows you to make this overlying structure transparent when performing beam plots.

This option is preserved for backwards compatibility. A more flexible method would be to adjust the visual transparency of the overlying structure, using the "quick pick" option, or explicit settings in the **PROPs** panel.

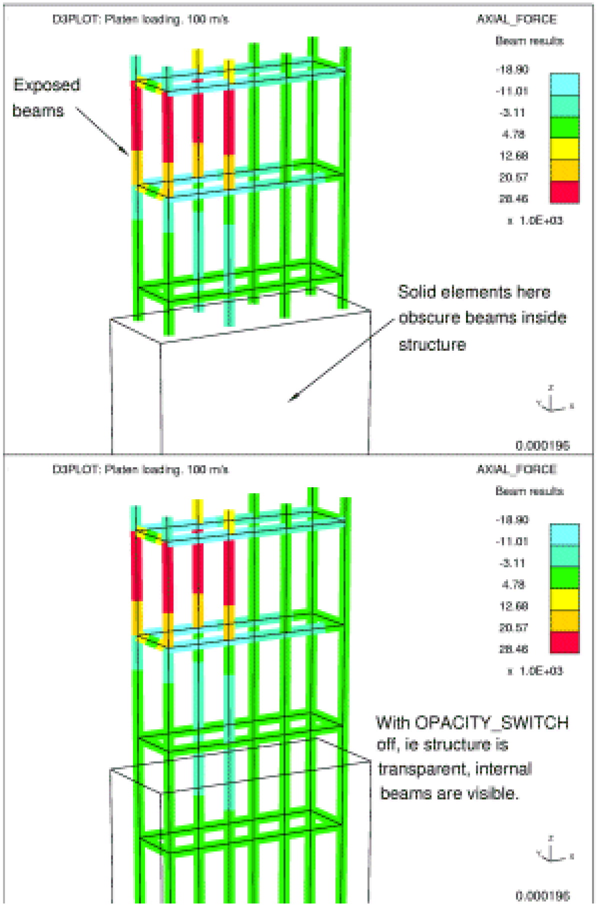
This example shows the affect of the Opacity Switch on a Beam data component.

Normally overlying structure will obscure (correctly) any beams that are behind it.

This example shows a typical concrete column with reinforcement and, clearly, it would be useful to visualise forces in the enclosed area while still seeing the external concrete outline.



When the **OPACITY_SWITCH** is turned off the overlying structure is no longer opaque, (i.e. it becomes transparent), and the results in the beams obscured above become visible.

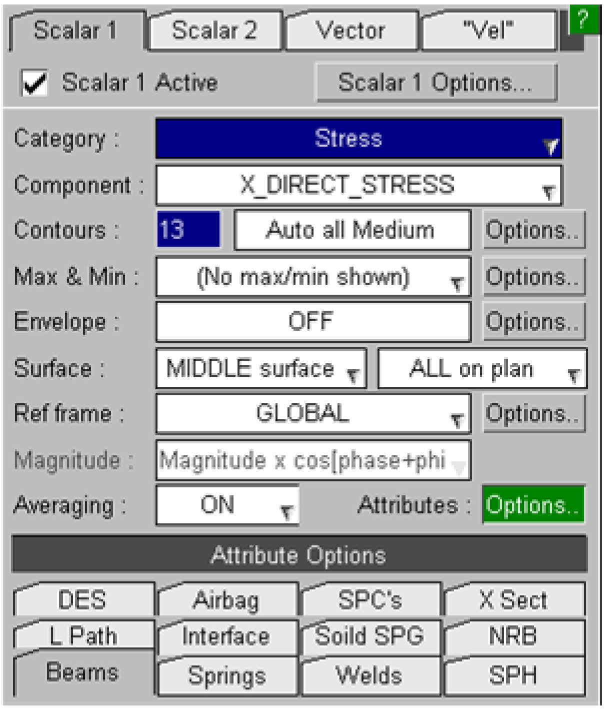


By default the **OPACITY_SWITCH** is off, i.e. beams which should not be visible are indeed obscured by the overlying structure. It can be turned on/off at will.

Note: The **OPACITY_SWITCH** affects both beam and contact surface data plotting modes (the same switch in both contexts). It has no effect in other contexts, or upon plotting modes that do not display data.

7.4.11. Attributes

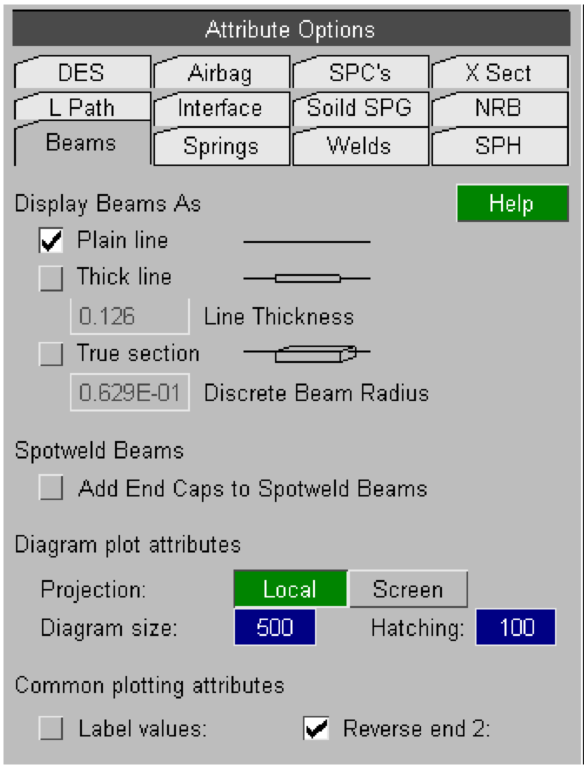
Attributes



Beams

These options control how beam elements are drawn and how data is plotted in Diagram Plots.

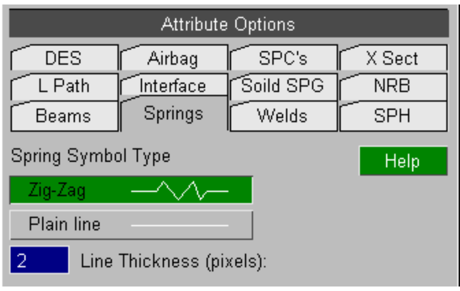
See BEAM_SYMBOLS... menu: Setting the drawing style for beams for more details.



Springs

These options control how spring elements are drawn.

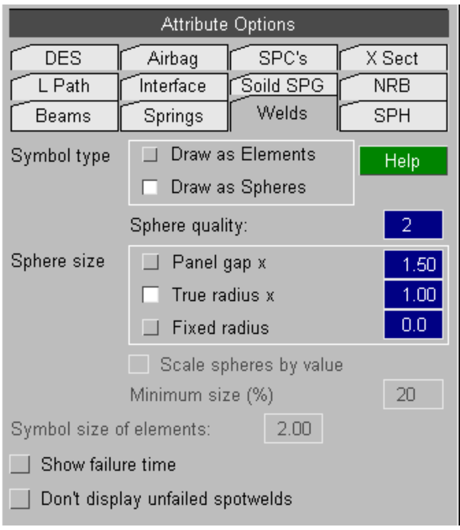
See SPRING_SYMBOLS... Menu: Setting the Drawing Style for Springs and Dampers for more details.



Welds

These options control how spotwelds are drawn.

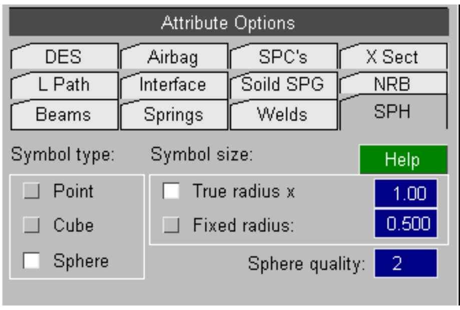
See Spotweld Symbols: Managing Spotweld Element Display for more details.



SPH

These options control how SPH elements are drawn.

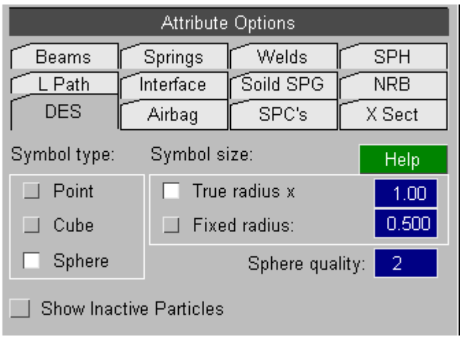
See SPH Symbols Managing SPH Element Display for more details.



DES

These options control how DES elements are drawn.

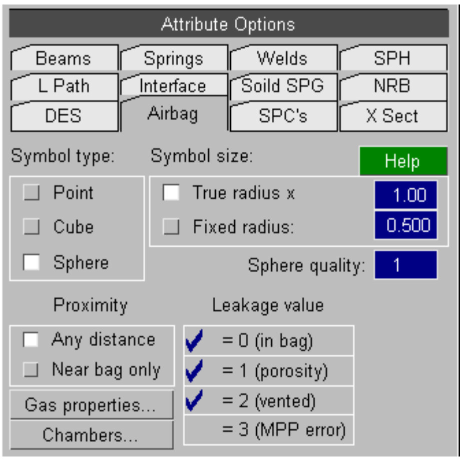
See DES Symbols for more details.



Airbag

These options control how Airbag particles are drawn.

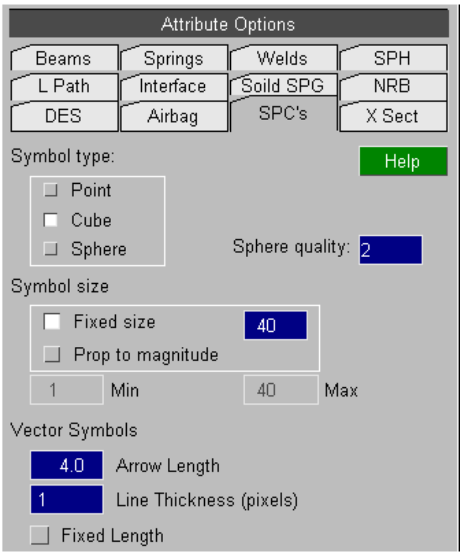
See AB Pcle Symbols: Managing Airbag Particle Display for more details.



SPC's

These options control how SPC's are drawn.

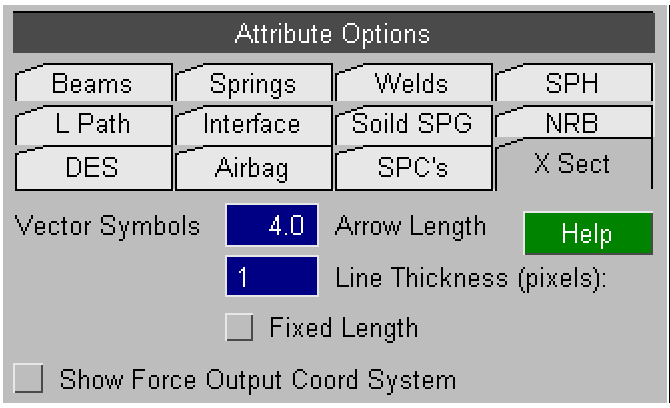
See SPC Symbols for more details.



X Sect

These options control how X Sections (*DATABASE_CROSS_SECTION) are drawn.

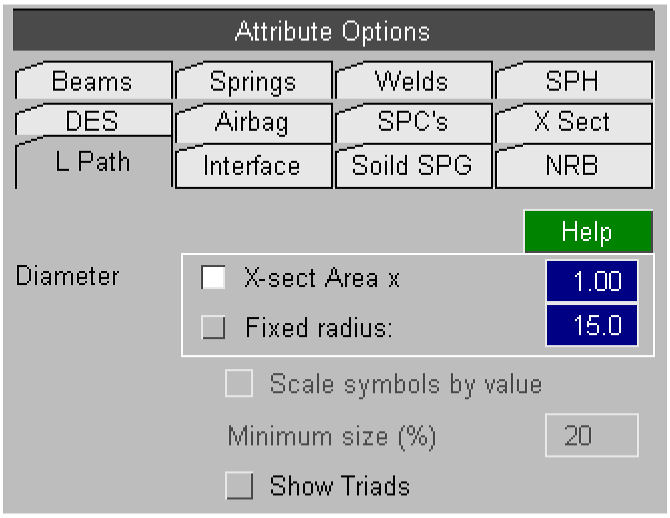
See X-Section Symbols for more details.



L Path

These options control how Load Paths are drawn.

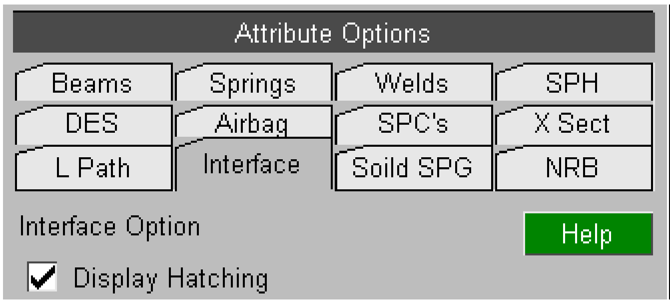
See Load Paths for more details.



Interface

These options control how Interface (contact) segments are drawn.

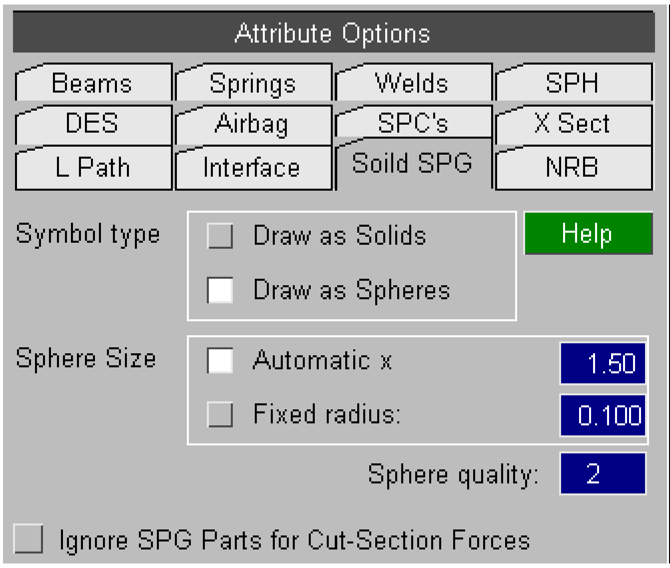
See Interface Symbols for more details.



Solid SPG

These options control how Solid SPG Parts (*SECTION_SOLID_SPG) are drawn.

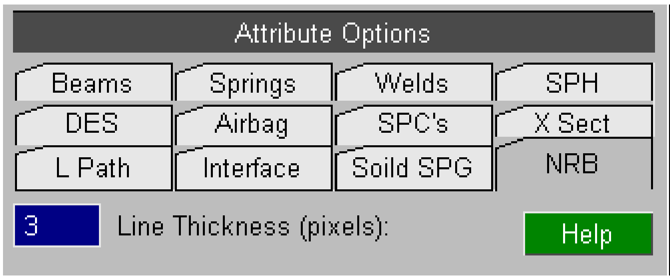
See Solid SPG Symbols for more details.



NRB

These options control how NRB's (Nodal Rigid Bodies) are drawn.

See Interface Symbols for more details.



7.5. DATA COMPONENTS - ADVANCED

DATA COMPONENTS - ADVANCED

From D3PLOT 13.0 onwards, D3PLOT can plot multiple data components.

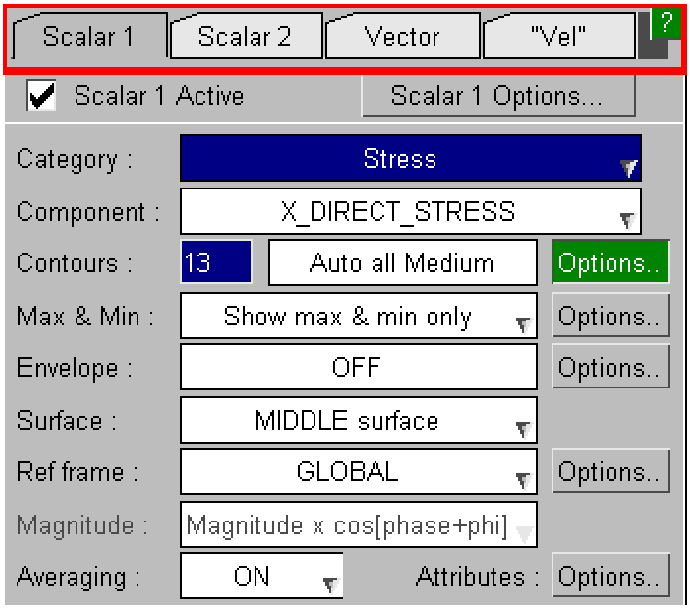
The tabs at the top of the **DATA COMPONENT** menu can be used to setup 4 different data components.

Scalar 1

Scalar 2

Vector

Vel



7.5.1. "Scalar 1" and "Scalar 2" Components

"Scalar 1" and "Scalar 2" Components

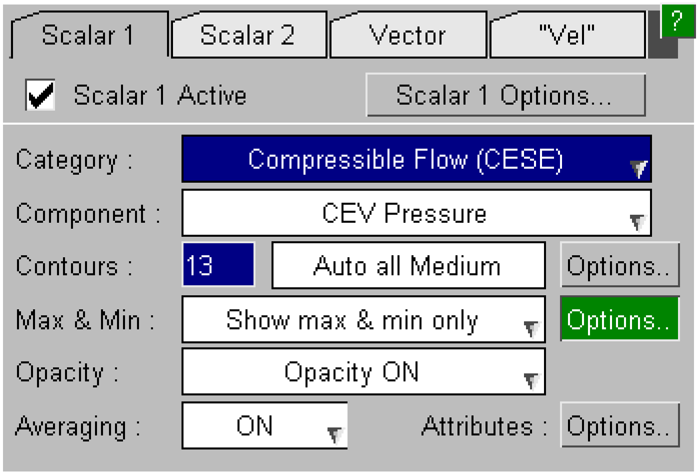
The **Scalar 1** and **Scalar 2** tabs can be used to set up 2 different data components that can be plotted in CT, SI, LC, CL or ISO plotting modes (see Drawing commands that plot data).

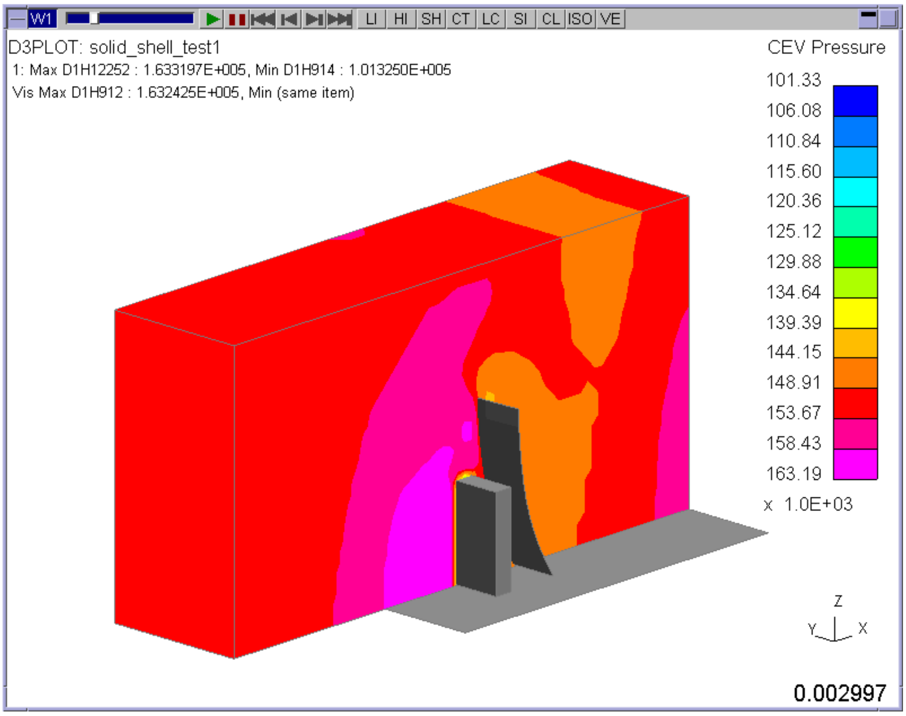
Components are plotted by entity type so one component can be plotted for some entity types (e.g solids and shells) and a 2nd component can be plotted for another entity type. Different components cant be plotted for different parts of the same type so you cant have half the shells in a model displaying one component and half a different component.

A single element can only display 1 data value at a time. If both the **Scalar 1** and **Scalar 2** data components are defined and are valid for the same entity type then the 2nd component is used. If however the components are valid for different element types then both components will be displayed and contoured at the same time.

The **Options** button (see "Options") can be used to give more control over which component is plotted on each entity type if both **Scalar 1** and **Scalar 2** are valid for the same entity types.

Setting **Scalar 1** to CESE Volume Pressure.

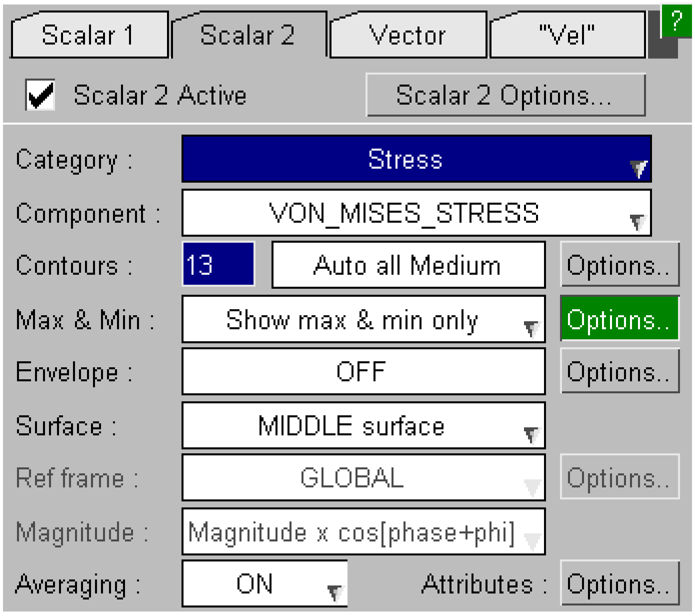


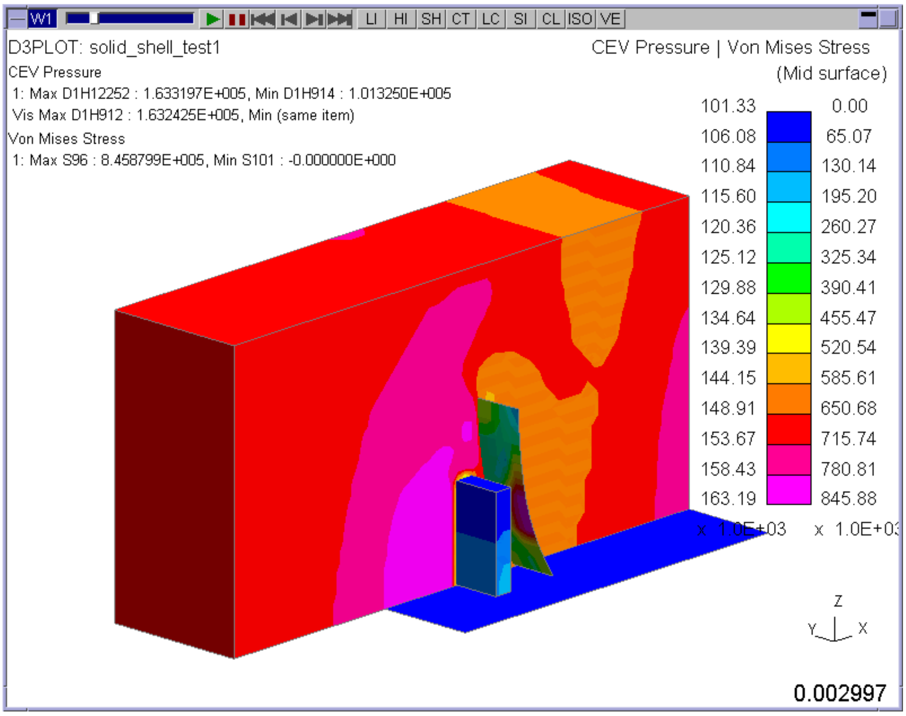


Pressure on CESE Volume

To setup a 2nd data component select the **Scalar 2** tab, select the component and tick the **Scalar 2 Active** option so that it is used.

If for example the 2nd component is set to Von-Mises Stress which is valid for structural elements and the component is set to be Active.



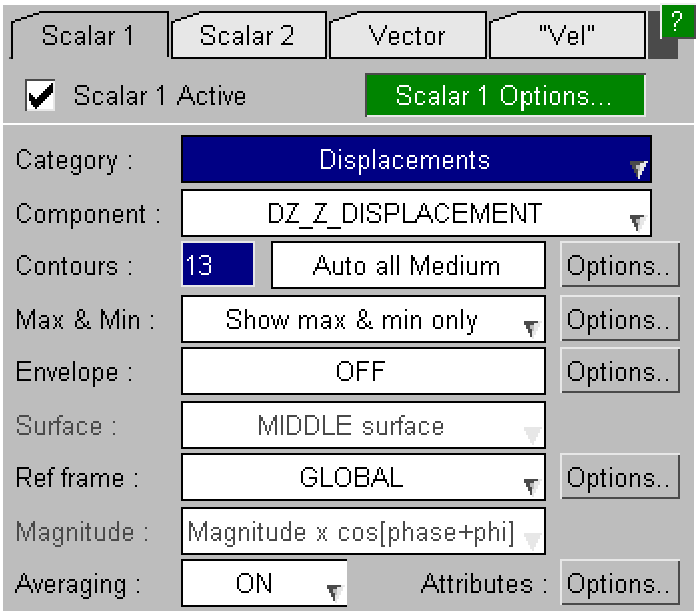


Pressure on CESE Volume + Von Mises Stress on Structural Elements

7.5.2. "Options"

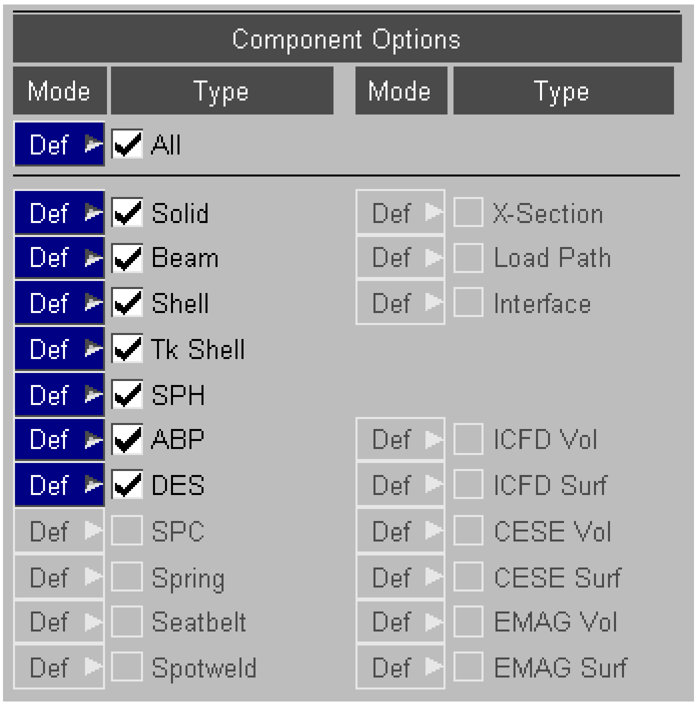
"Options"

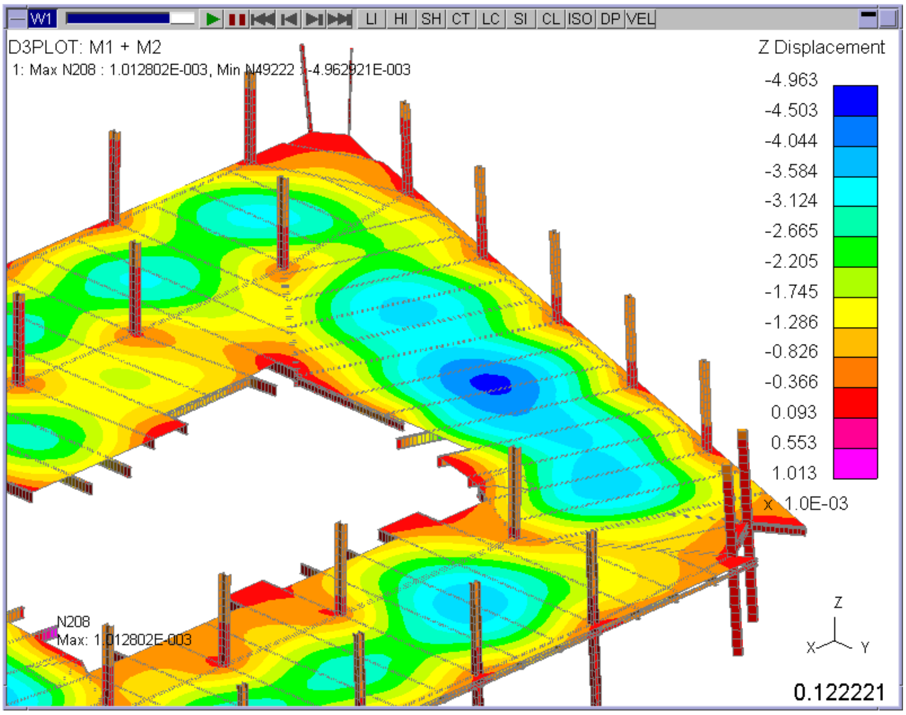
For each data component D3PLOT knows what entity types it is valid for and can be displayed on. By default a component is show on all the valid entity types but this can be changed using the **Options...** button.



If for example **Scalar 1** is set to the component "Z Displacement" then by default it is contoured on the entity types shown opposite.

Any entity types that the component can not be contoured on are automatically greyed-out and can not be selected/de-selected.

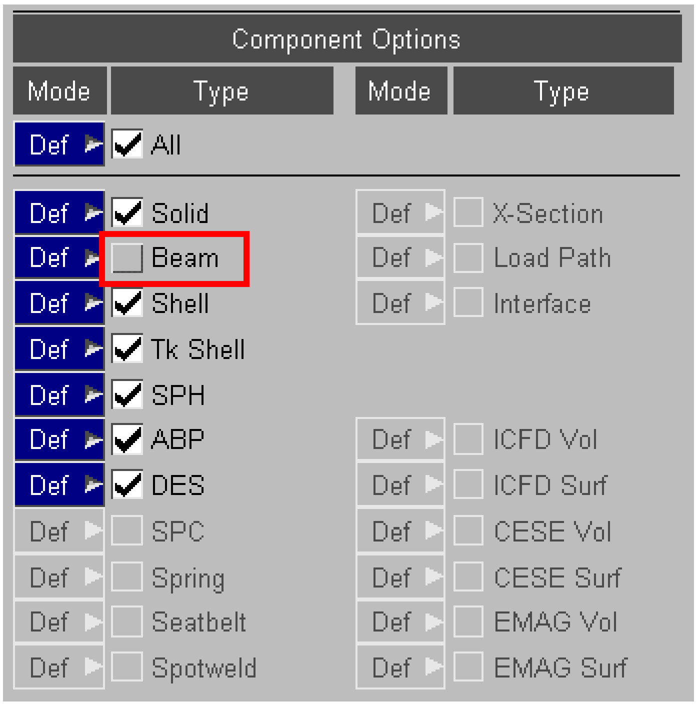


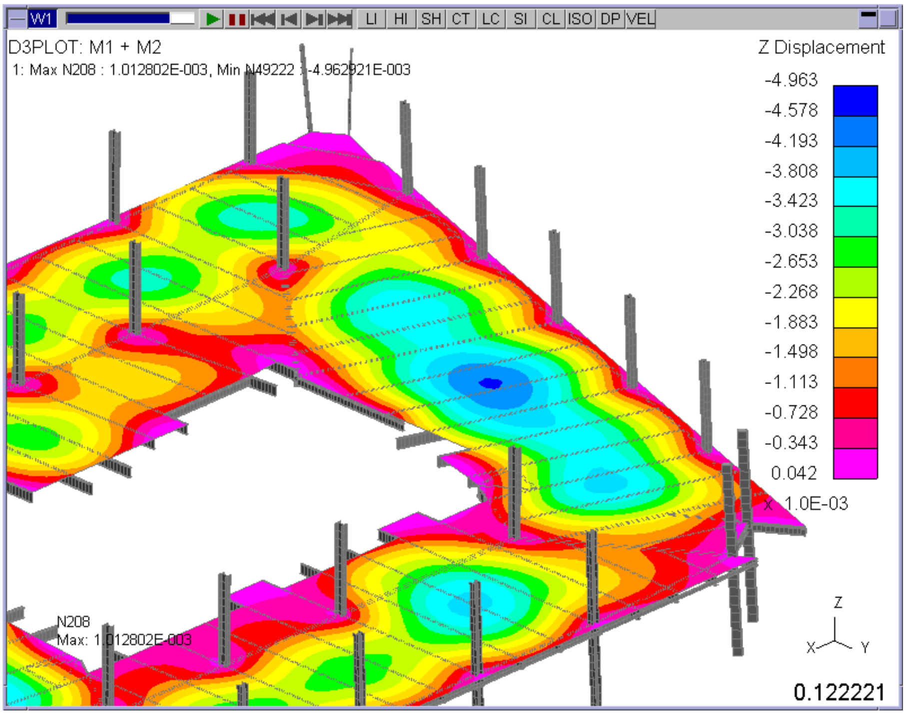


Z Displacement - Default entity types

To exclude an entity type from being used by a data component deselect it from the list of selected types.

If for example the Z Displacement of the beam elements was to be excluded.





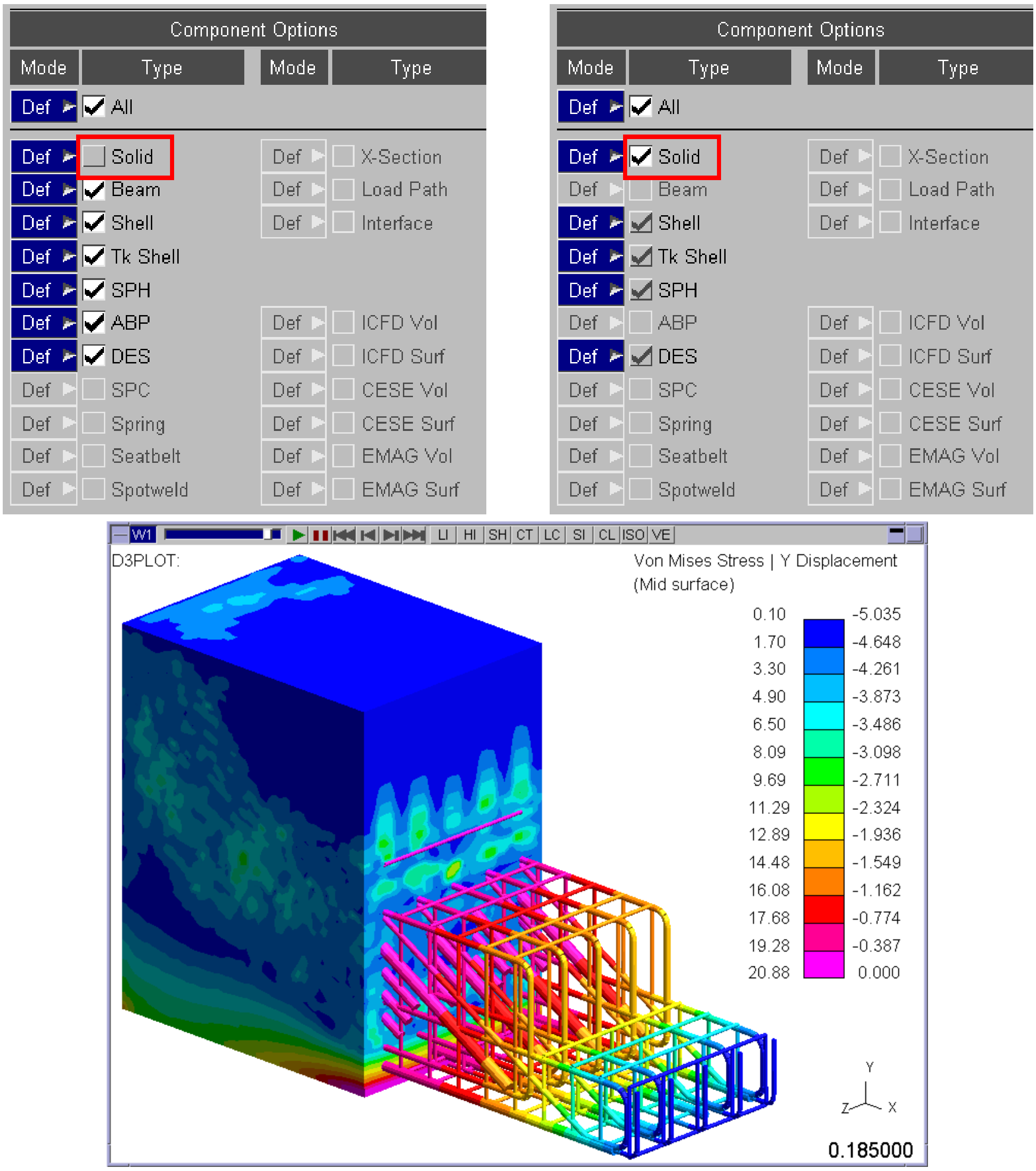
Z Displacement - Beams Excluded

If both **Scalar 1** and **Scalar 2** are valid for the same entity types then by default **Scalar 2** would be displayed on all the entities.

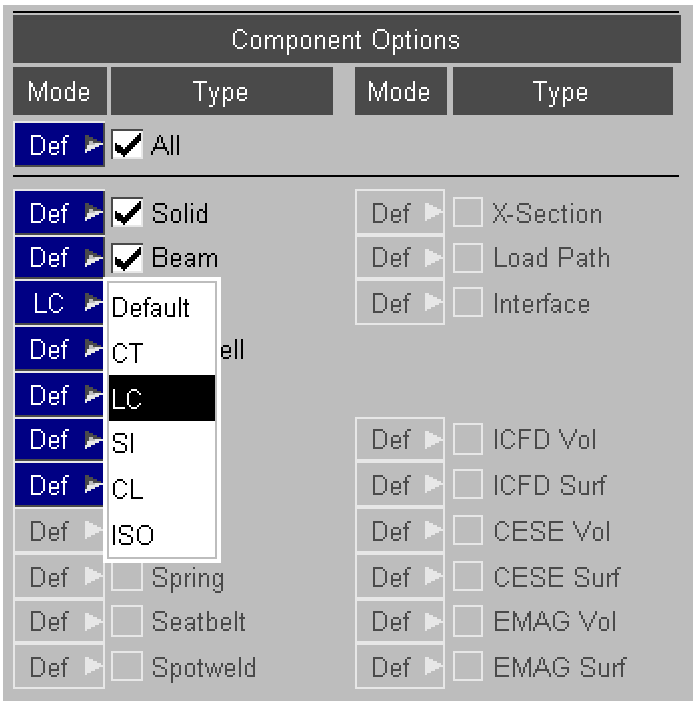
The **Options** menu can be used to turn **Scalar 2** off for some entity types so that on those entity types **Scalar 1** is displayed.

Setting **Scalar 2** to Z Displacement and using the **Options** menu to turn Scalar 2 off for Solids.

Setting **Scalar 1** to Von-Mises Stress and using the **Options** menu to turn **Scalar 1** on for Solids

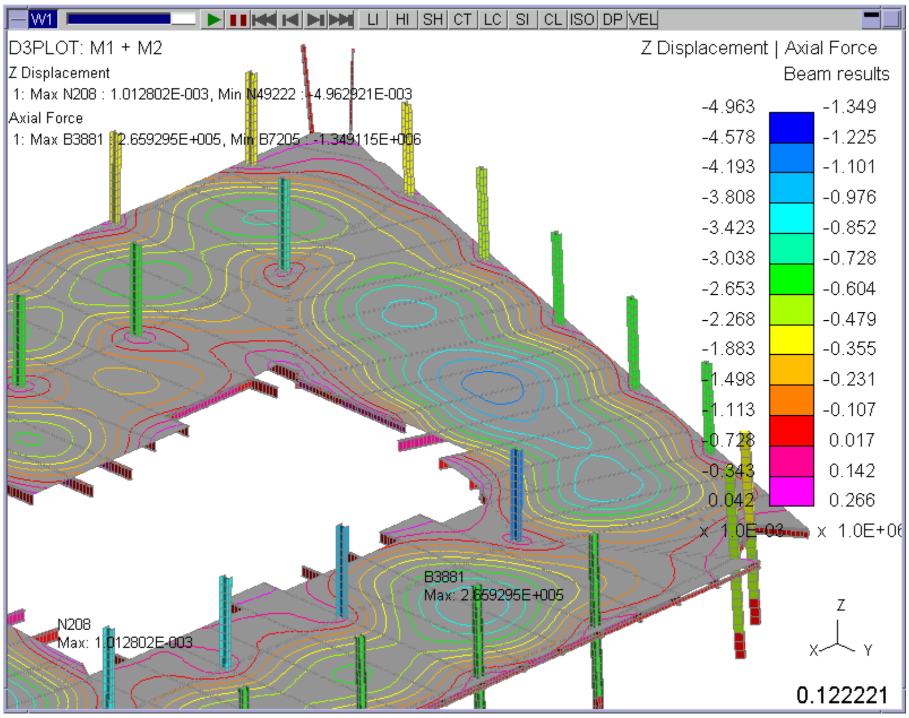


Von Mises Stress on Solids + Y Displacement on Beams



As well as turning on and off data components the **Options** menu can be used to modify the plotting mode used for different entity types. By default all entity types are contoured using the default plotting mode so an SI plot will contour everything in SI mode and a CL plot will contour everything in CL mode.

If however you wanted to generate a LC (line contour) plot with the beams drawn in SI you can change the shell plotting mode to LC and this would override the default plotting mode and then generate an SI plot.



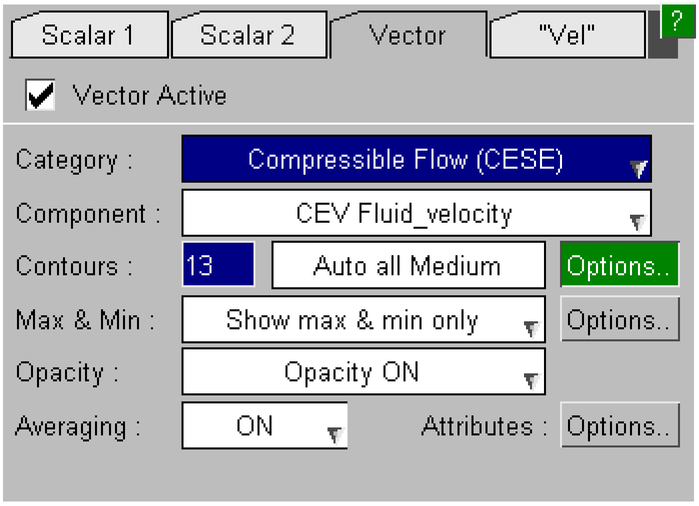
LC plot of Z Displacement "Scalar 1" + SI plot of Beam Axial Forces "Scalar 2"

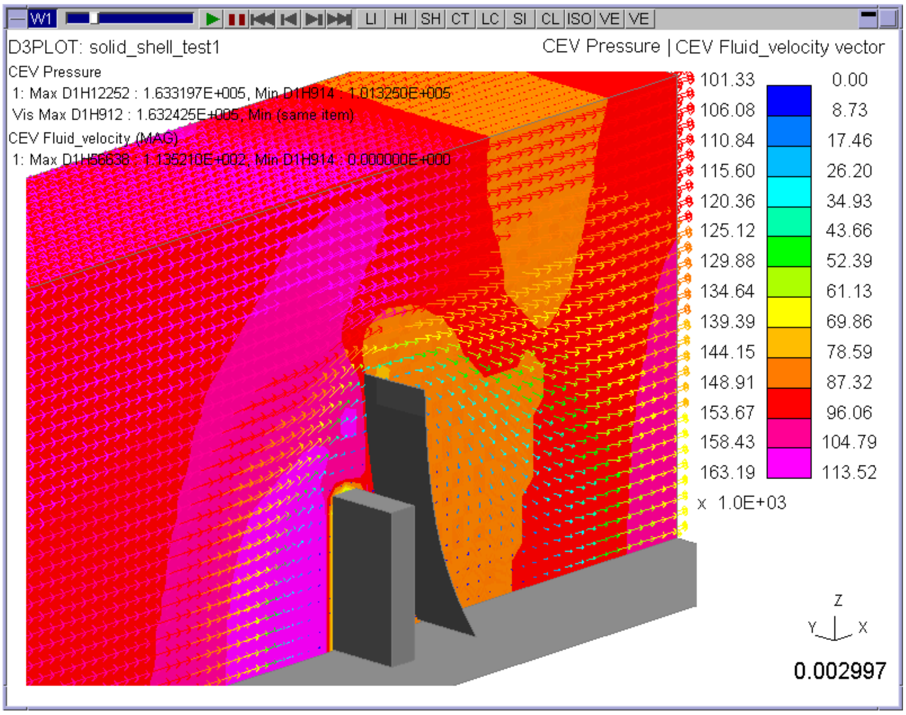
7.5.3. "Vector" Component

"Vector" Component

As well as simultaneously displaying 2 different Scalar components a separate **Vector** component can also be displayed.

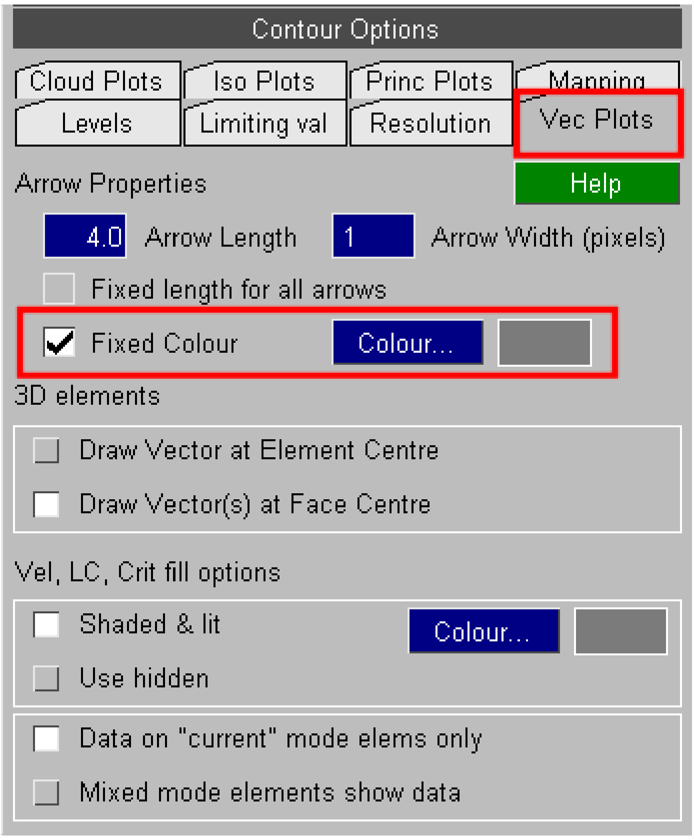
This can be displayed separately using the VEC plotting mode (see Drawing commands that plot data), or it can be combined and drawn on top of the **Scalar 1** and **Scalar 2** components.

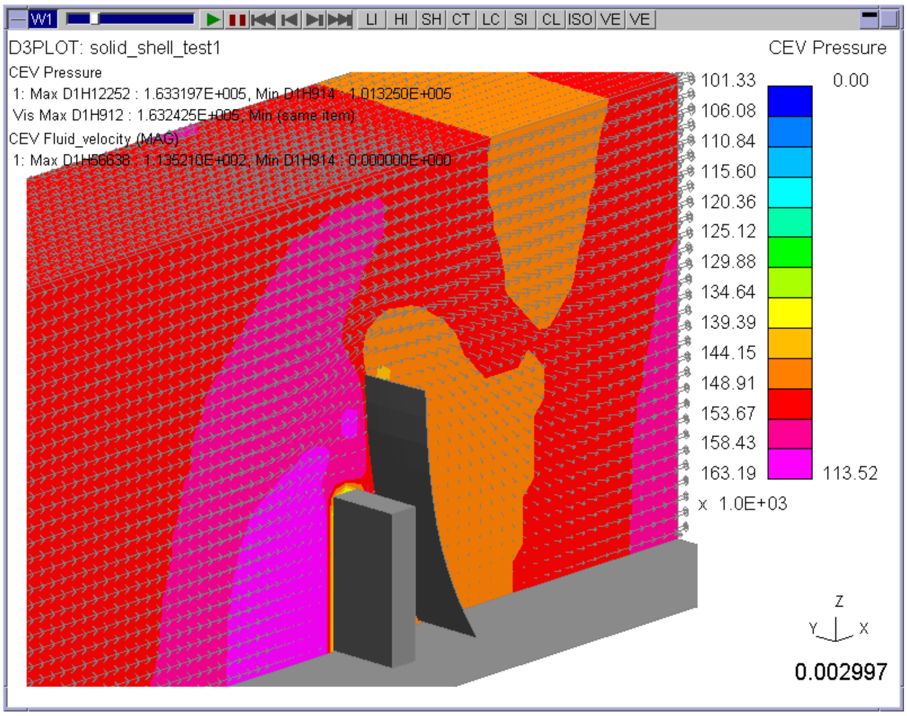




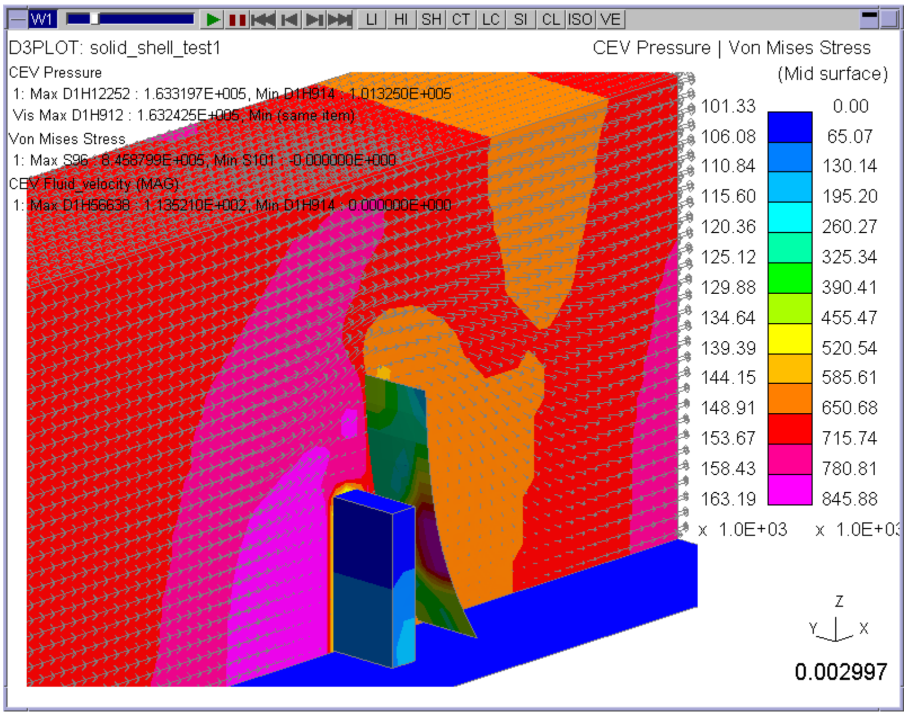
Pressure on CESE Volume + CESE Volume velocity vectors

To make it easier to see the flow pattern a single colour can be used for the Vector arrows where the magnitude is proportional to the arrow length. This option can be found under the **Vec Plots** tab in the Contour Options menu (see Vec Plots)





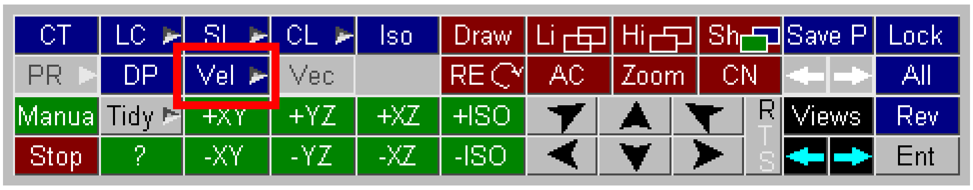
If all 3 data components are defined then only **Scalar 1** and **Scalar 2** are displayed on the contour bar and the **Vector** Arrows will automatically be drawn using a single colour.



7.5.4. "Vel" Component

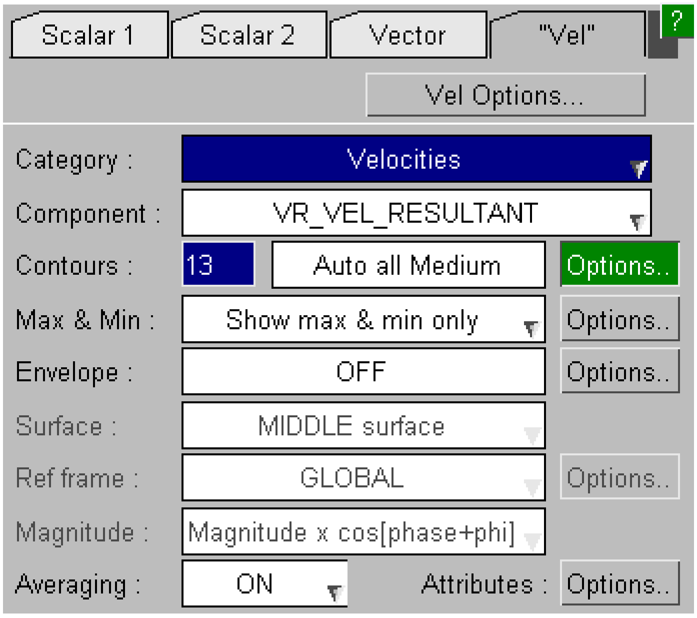
"Vel" Component

The final tab Vel is a special data component used by the **Vel** plotting mode.



It can only be set to either Velocity, Displacement or Acceleration.

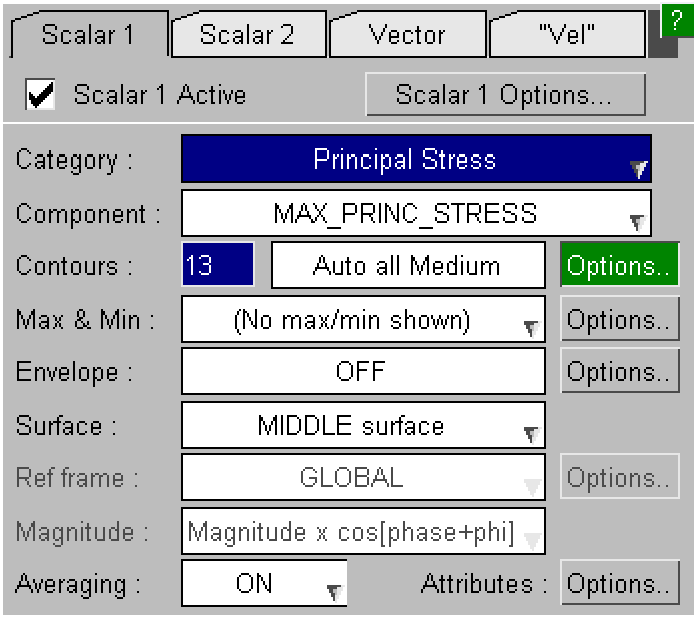
As velocity plots are a very common plot the **Vel** button can be used to generate a Velocity plot at any time regardless of the settings in the other data component tabs.

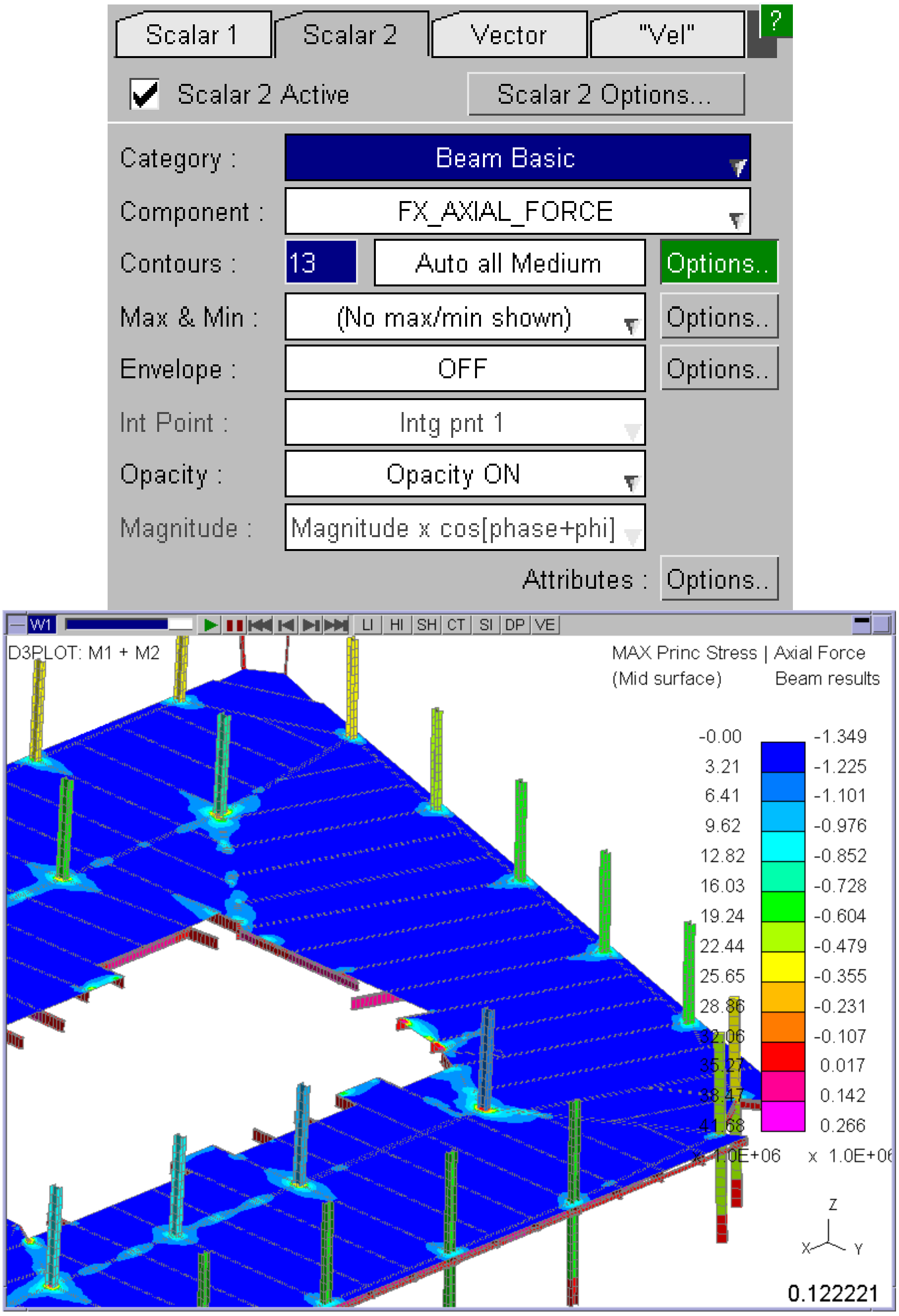


7.5.5. Invalid Data Components and Entity Types

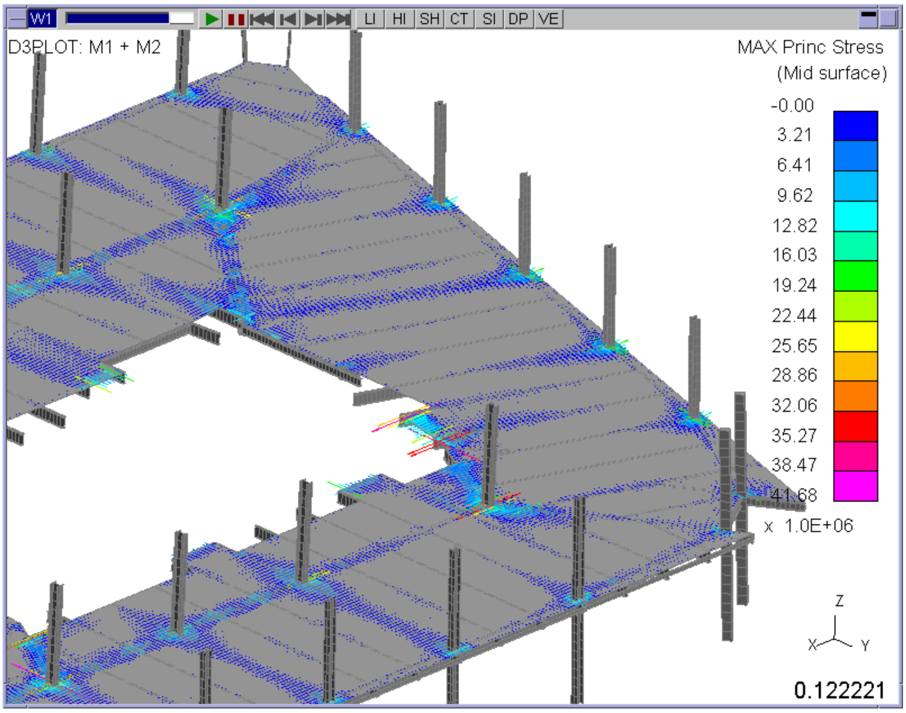
Invalid Data Components and Entity Types

If the Scalar 1 and Scalar 2 data components are defined and one of them is not valid for a plotting mode then that data component will automatically be omitted from the contour bar and min/max values.

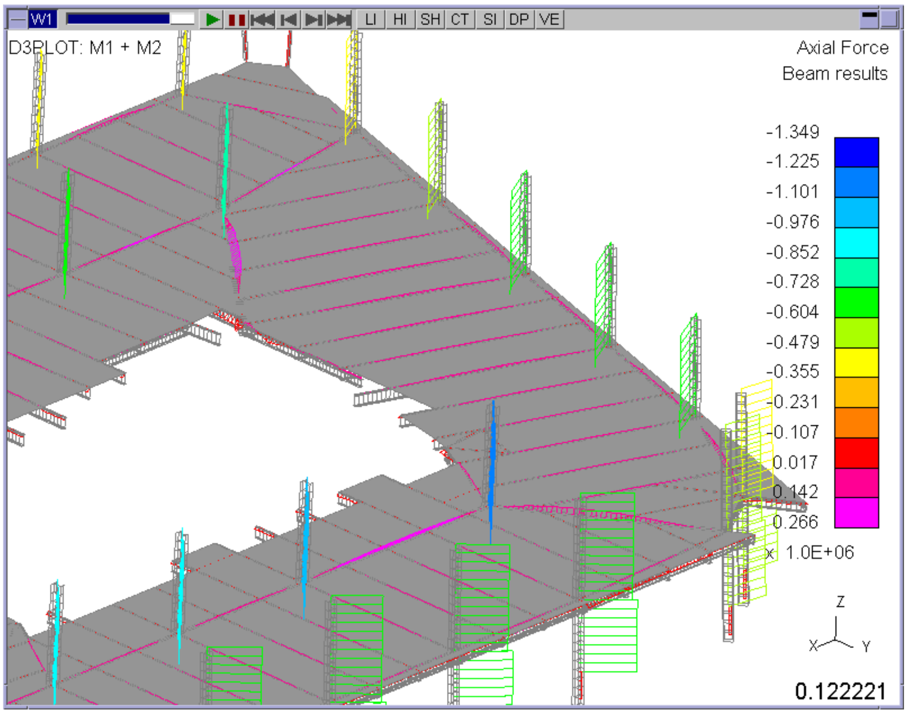




SI plot shows both Max Principal Stress + Beam Axial Force

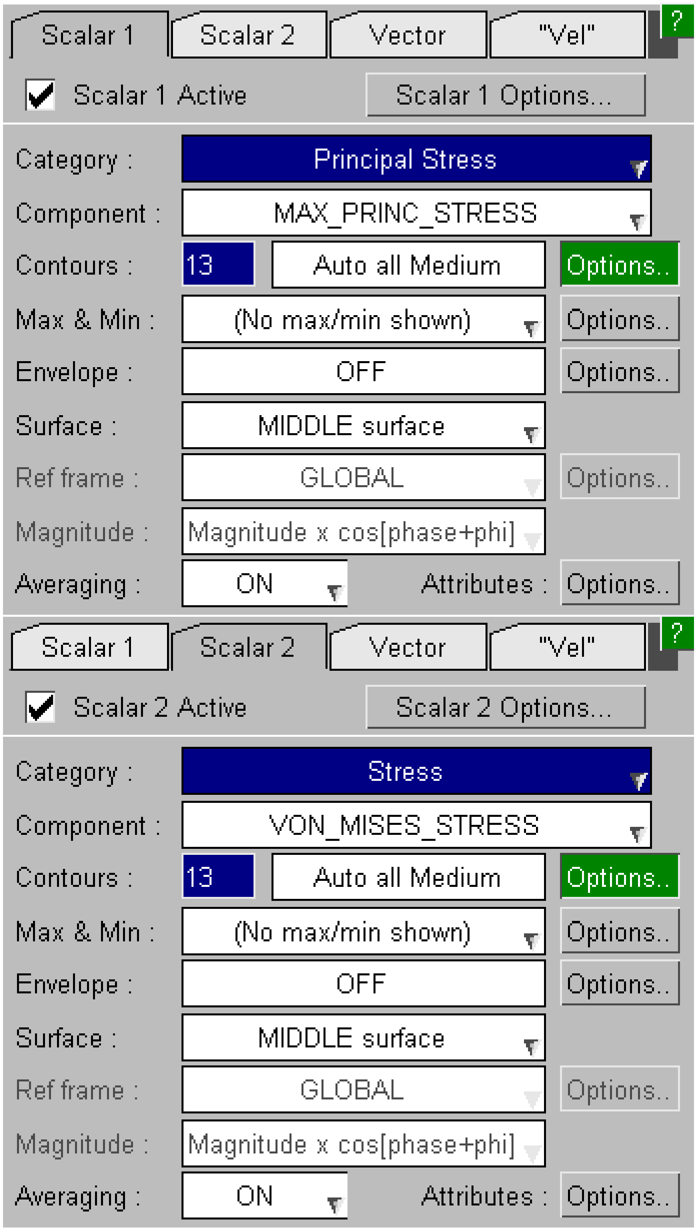


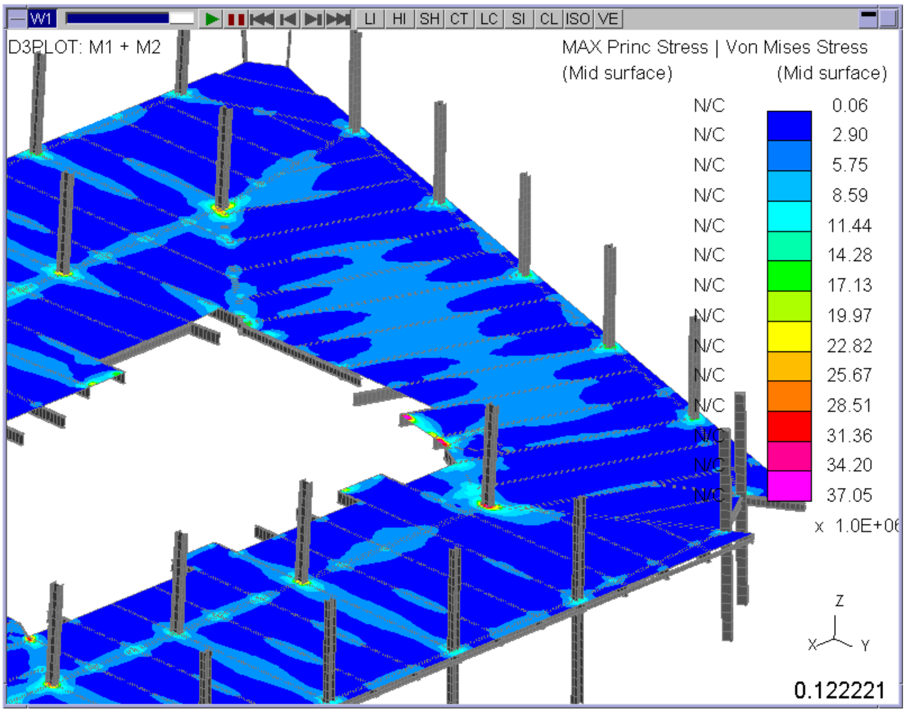
PR plot only show Max Principal Stress



DP plot only shows Beam Axial Force

If a data component is setup that is valid but no elements are drawn or contoured using that data component then the component will still be shown on the contour bar but the values will be displayed as N/C to show that nothing was computed for that component.

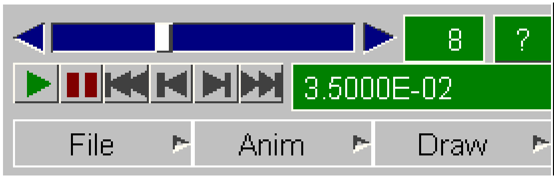

Scalar 1
Scalar 2
Vector
"Vel"
?
Scalar 1 Active
Scalar 1 Options...
Category :
Principal Stress
Component :
MAX_PRINC_STRESS
Contours :
13
Auto all Medium
Options..
Max & Min :
(No max/min shown)
Options..
Envelope :
OFF
Options..
Surface :
MIDDLE surface
Ref frame :
GLOBAL
Options..
Magnitude :
Magnitude x cos[phase+phi]
Averaging :
ON
Attributes :
Options..
Scalar 1
Scalar 2
Vector
"Vel"
?
Scalar 2 Active
Scalar 2 Options...
Category :
Stress
Component :
VON_MISES_STRESS
Contours :
13
Auto all Medium
Options..
Max & Min :
(No max/min shown)
Options..
Envelope :
OFF
Options..
Surface :
MIDDLE surface
Ref frame :
GLOBAL
Options..
Magnitude :
Magnitude x cos[phase+phi]
Averaging :
ON
Attributes :
Options..



SI plot, N/C shown for Scalar 1 as all the shells are being contoured using Scalar 2 Von Mises Stress

7.6. Animation How to Display, Control, Store and Retrieve Animation Sequences

Animation How to display, control, store and retrieve animation sequences

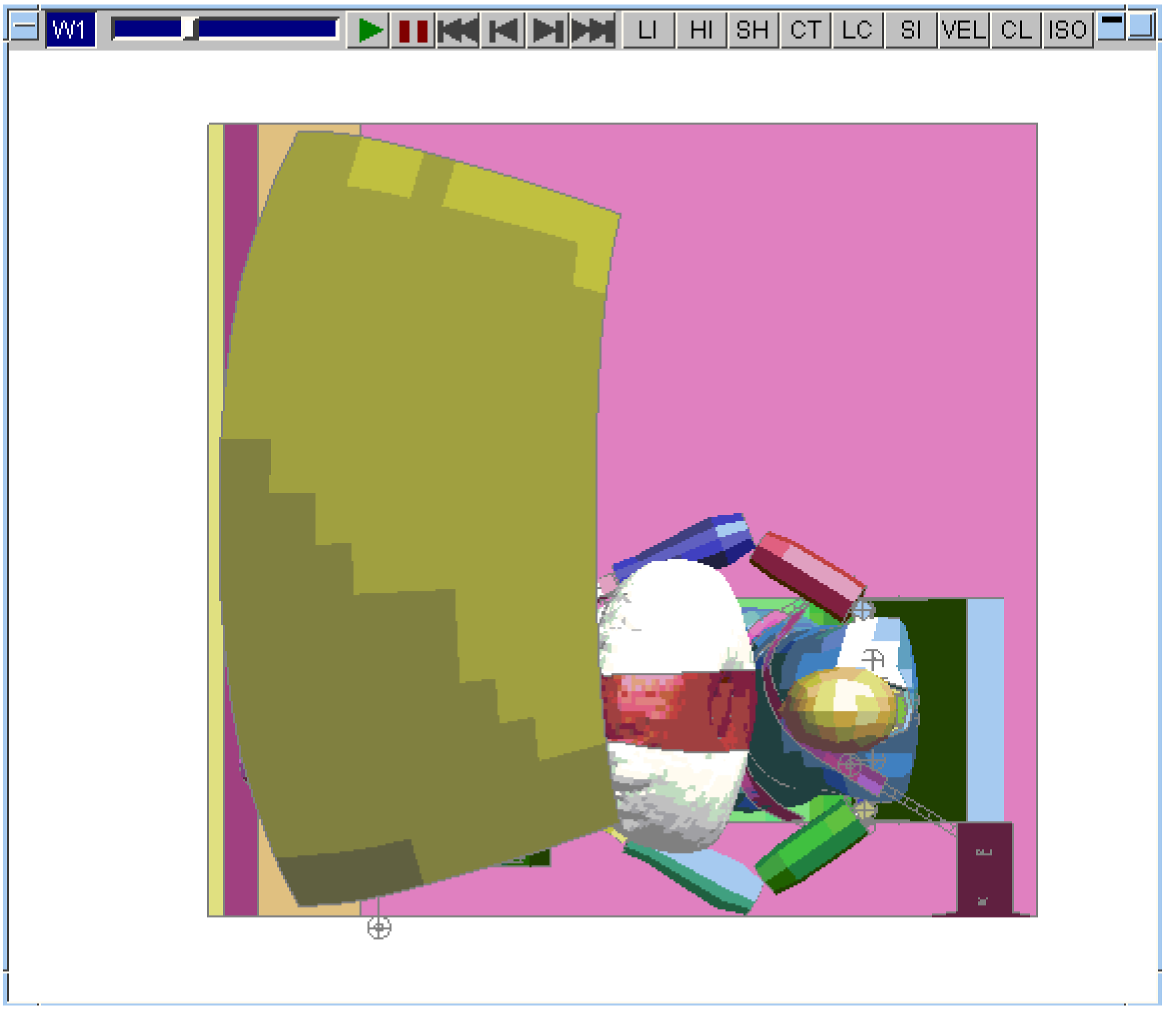


In D3PLOT 8.0 there is no distinction between "static" and "animation" modes: a static plot can be thought of as a frozen slice of a dynamic animation sequence. Any conventional display mode can be animated at any time by pressing **PLAY >** , and halted again with **STOP** .

Virtually all menu functions such as blanking, component change, and so on can be used while animating, as can dynamic viewing: it is not necessary to halt an animation in order to change the attributes of your image.

7.6.1. Basic Animation Controls

Basic animation controls



As well as the **PLAY >** and **STOP** controls in the **State Display** box the graphics window has a set of controls at its top left which can be used to control animation, and the state used for static display.

The master slider in the **State Display** panel controls all graphics windows for which its **W *n*** tabs are active.

The slider and associated controls at the top of a graphics window control that window only.

 **STOP** s the current animation.

 Starts animation (same as **PLAY >**)

 Jump straight to first or last frame 

 Single step backwards (**|<**) and forwards (**>|**) 

The state slider allows you to scroll with the mouse to any state, and to slide dynamically through them.

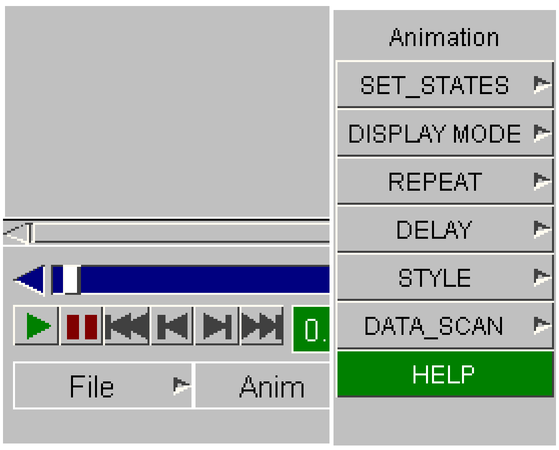
If an animation is playing using any of these controls (other than **PLAY**) will automatically stop it in order to execute the new commands.

7.6.2. ANIM > Controlling the Animation Process

ANIM > Controlling the animation process

The **ANIM >** popup menu gives access to the controls for animation.

Each option has a set of standard sub-menus, and further "custom" defaults for more complex settings.



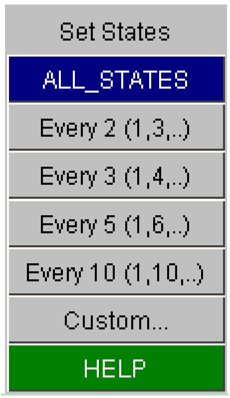
7.6.2.1. SET_STATES > Selecting the States to Be Animated

SET_STATES > Selecting the states to be animated

By default all states are selected for animation. You can select from the pre-programmed **Every n** options here, or use **Custom...** to define more precisely what is to be animated.

The **ALL_STATES** and **Every *nn*** options will apply to all windows with active **W *n*** tabs in this panel.

Where multiple models are present the **Custom...** states can only be selected for one model at a time. You should set the **W *n*** tabs to the window(s) of this model first: attempting to select custom states for multiple models will generate warnings.



Interpolating animations by time

Normally animations are drawn at the states defined in your database(s), but it is possible to interpolate by time between states: both to get a smoother progression through a sequence with too few states, and also to match multiple models with dissimilar output frequencies. This is described under "custom" animation **BY_TIME...** below.

Animating Static and Eigenvalue (modeshape) analyses

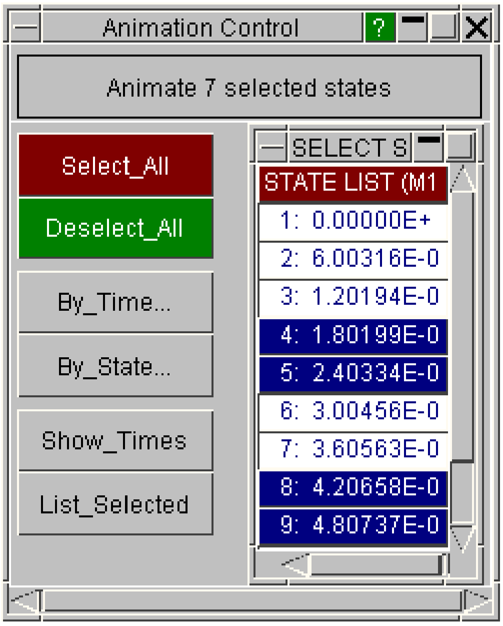
This section assumes the normal case of a transient analysis producing a series of states at successive times. However it is also possible to animate a single static or eigenvalue state by cycling it through 0 - 360 degrees. This is described in Animating static and eigenvalue (modal) analyses.

Custom animation definition

You can select any permutation of states from the **STATE LIST** menu. (In this example states 7, 8, 11, 12 have been de-selected.)

(DE-)SELECT_ALL	(De-)selects all states in the **STATE LIST** menu.
SHOW_TIMES	Lists all available states in the database file.
LIST_SELECTED	Lists more details about the currently selected states.
BY_STATE...	Allows you to select states via <start> <increment> <end> syntax. (Convenient for models with very many states.)
BY_TIME ...	Interpolating between states by defining time intervals.

Note that this "custom" panel can only apply to one model at a time. When multiple models are present it will be restricted to the window(s) of a single model.



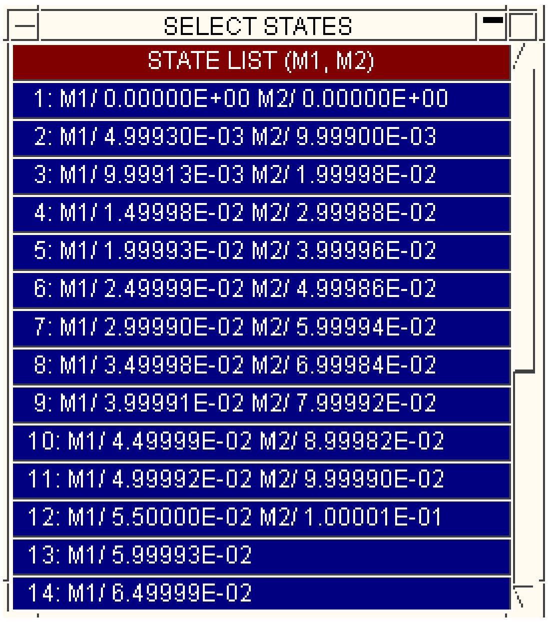
How models with dissimilar states are animated

The examples above consider the case of only one model. Where two or more models exist it is possible that they will have a different number of states, at different time intervals.

Consider the following example of two models with dissimilar states:

Model 1 states	Model 2 states
1. T = 0.000	1. T = 0.000
2. T = 0.100	2. T = 0.200
3. T = 0.200	3. T = 0.400
4. T = 0.300	4. T = 0.600
5. T = 0.400	5. T = 0.800
6. T = 0.500	6. T = 1.000

The **SELECT STATES** menu in this example will now show how the states align by frame, as follows:



7. T = 0.600 8. T = 0.700 9. T = 0.800 10. T = 0.900 11. T = 1.000	

Animation sequence when models have dissimilar numbers of states

In this situation a model with fewer states waits until the one with more states has finished its sequence before looping back to zero. Using the example models above we now get:

Frame	Model 1 shows	Model 2 shows
1.	T = 0.000	T = 0.000
2.	T = 0.100	T = 0.200
3.	T = 0.300	T = 0.400
4.	T = 0.400	T = 0.600
5.	T = 0.500	T = 0.800
6.	T = 0.600	T = 1.000
7.	T = 0.700	T = 1.000 < = Hold last state
8.	T = 0.800	T = 1.000 < = Hold last state
9.	T = 0.900	T = 1.000 < = Hold last state
10.	T = 1.000	T = 1.000 < = Hold last state
11.	T = 0.000 **<= Loop back to state 1**	T = 0.000 **< = Loop back to state 1**
12.	T = 0.100	T = 0.200
13.	T = 0.200	T = 0.400

Note that the "holding" operation in M2 is based on state number, not time.

This "hold last state" logic applies whether the dissimilar models are in the same or different windows. (This is a change of behaviour in D3PLOT V92, in previous versions there was no synchronisation between windows, and models were only "held" if they were in the same window.)

What the "clock" in a graphics window shows with two or more models in a window

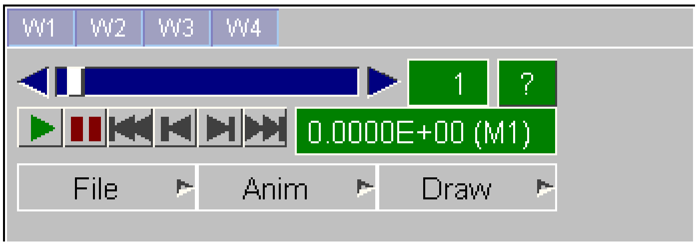
Where there is only one model in a window then there is no ambiguity, and the clock at the bottom right shows the current state's time.

Where there are two or more models in a window then the clock shows:

- **`max (Time of M1, Time of M2, ...)`**
- If the times in the multiple models do not match within 0.1% then the clock's colour is inverted, typically black on white.

Synchronising animations by state across multiple windows

From D3PLOT 9.2 onwards, animation in multiple windows is aligned by state number. Thus all animations will start at state #1, and step forwards together through states #2, #3, .. #n. If a window has fewer states than one or more other windows it will wait at its last state until the window with the highest number of states has reached its end, then they will all loop back to state #1 together.



The **STATE NUMBER** slider will set the selected state for all active windows (here **`W1 .. W4`**), stopping animation if it is currently running.

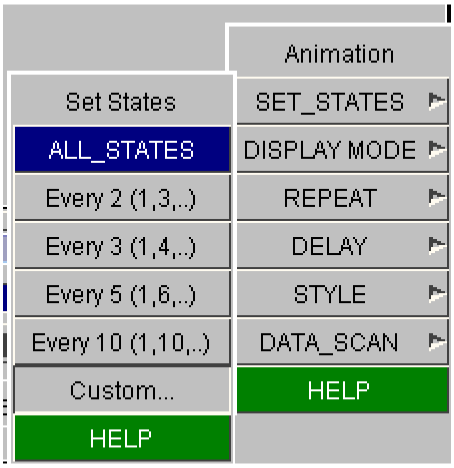
If you subsequently restart animation with the master **PLAY >** button in this panel then all windows will commence from the same state and, if they have the same number of frames, will remain synchronised.

Synchronising animations by state when windows have different numbers of states

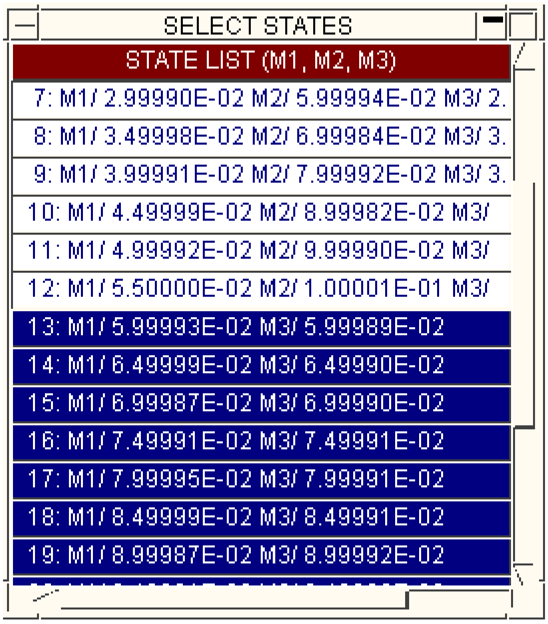
If they do not have the same number of states, or their states have different time intervals, animations across multiple windows may fall out of step in terms of "true" analysis time.

The remedy is to animate by explicit time interval, rather than by state number. This is done by interpolating "By Time" as follows:

Use **ANIM > SET_STATES > Custom...** to map the detailed state selection panel:

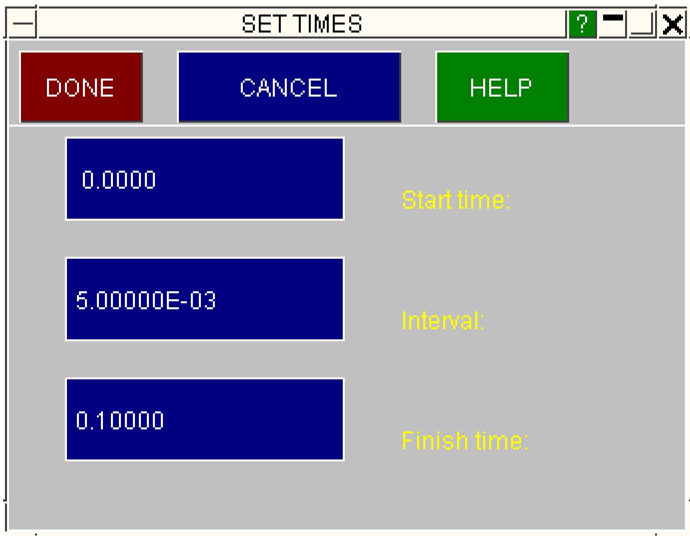


Then truncate the number of states in all windows so that they all have the same number of states. In this example it has been limited to 12states.



BY_TIME... Synchronising animation by time across dissimilar models

The "by state" animation behaviour above is the default, chosen because it is simple and minimises memory usage, but it has disadvantages when models with different state time intervals are processed.



It is possible to interpolate between states and so to animate by user-defined time interval, which has the side-effect of synchonising models with dissimilar state times. This is done by using the **BY_TIME...** option. (Interpolation can be used on single models as well, usually to give a smoother animation, although this is less common.)

You define

```
<start time>
<time interval>
<end time>
```

And D3PLOT will interpolate states as required to achieve the specified intervals.

Interpolation is performed using a simple linear factor on the two "real" states that bound the required time.

You can control animation on a per frame basis using the slider and associated controls at the top of the window, exactly as described in Basic Animation Controls above.

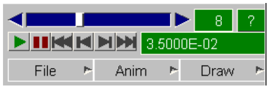
Controlling the display (statically) "by state".

To display the results (statically) at a "true" state while interpolating animations by time you need to revert to selecting the required state in the state control box.

This does not change animation back to "by state" mode, it simply reverts temporarily to showing the image at the time selected.

To revert to controlling animation by state, rather than by interpolated time, use any of the methods above to select states (eg **ANIM > SET_STATES**)

During interpolated animation by "time" the state box controls will always allow you to revert to showing a "true" state, not an interpolated one.



How interpolation by time affects output elsewhere in D3PLOT

Interpolation by time affects the following other parts of the code:

- **WRITE** output will be given at the currently interpolated time.
- **DEFORM** options that use nodal coordinates (**SHIFT_DEFORMED, FIX_NODE, REFERENCE_NODE(s)**) will use the interpolated coordinates of the relevant nodes.
- **CUT_SECTIONs** that follow nodes will likewise use the interpolated coordinates.
- **UTILITIES, MEASURE** by node will report geometry from interpolated nodal coordinates.

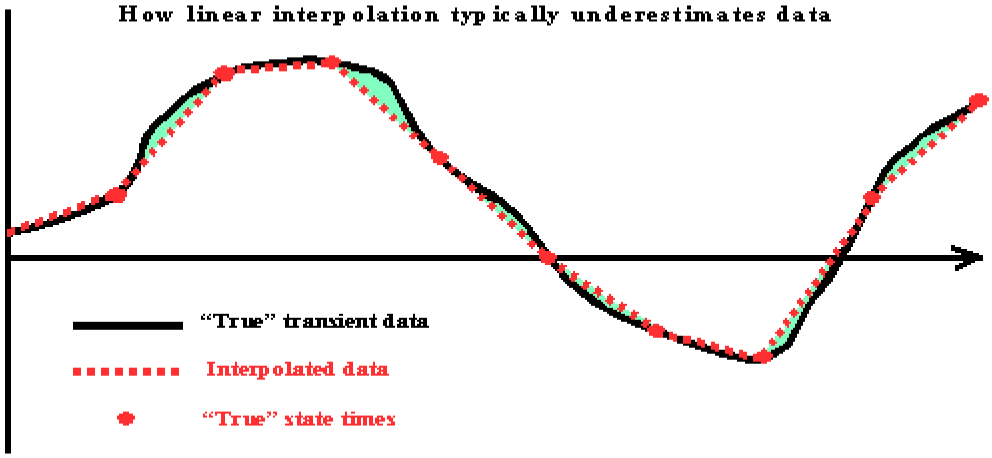
However note that interpolation by time does not affect the following:

- **XY_DATA** will still only report points at the time values of "true" states.

Warning: Interpolated values should be treated with care

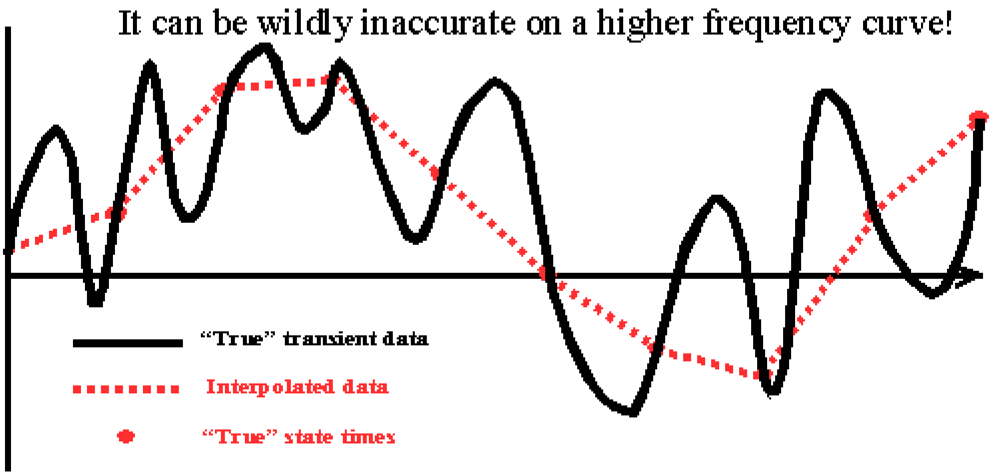
Linear interpolation tends to underestimate values between points on a smoothly varying, low frequency curve, and to give wildly inaccurate results on a curve which is high frequency relative to its sampling interval. This is acceptable for visual purposes during animation, but should be treated with caution if "written" values are to be extracted for subsequent use.

This is demonstrated by the following two plots:



Interpolation of a low frequency curve

In this case the green shaded areas show how the linearly interpolated results (red) underestimate the true values (black).



Interpolation of a high frequency curve

In this case, where the frequency of the curve is high relative to the sampling interval it is clear that linear interpolation (as well as the original points) will miss significant values.

Warning: Interpolation can be memory intensive

Interpolation by time requires more memory (to hold the interpolated data for display) which, combined with the memory required to hold multiple models, can result in machine memory becoming exhausted. It is better to make sure that models being compared have near identical output intervals, as this greatly simplifies post-processing.

Animation "frames" (as distinct from states)

The situation can arise in which the images to be animated are not explicit "states" in the database. This happens when:

- Transient analyses are animated at interpolated times. (eg **BY_TIME...** as described above)
- A single loadcase of a static (ex-Nastran) analysis is animated in modeshape form.
- An eigenvalue (ex-Nike/Dyna or Nastran) modeshape is animated.

(For more on static and eigenvalue animation see Animating Static and Eigenvalue (Modal) Analyses)

To handle this D3PLOT has the concept of animation "frames":

- Each image in an animation is a frame, regardless of its origin.
- In the normal (not interpolated) transient case there is an exact equivalence between animated states and frames.
- In other (interpolated, modeshape) cases there will be at least two frames in an animation, but usually many more.

How does this affect you? Not very much, you only need to know the following:

- The slider and other positioning controls at the top of the graphics window operate on "frames" not explicit states, although in most cases these are the same.

  Therefore in cases where you are animating something other than **ALL_STATES** the effects of scrolling this slider (which navigates frames) and the state slider in the **State Display** box (which navigates states) will be different.

- You cannot stop an animation at a frame that is interpolated between explicit states.

  D3PLOT will not permit you to operate statically on interpolated frames from a transient analysis. This is because such results are potentially misleading: linear interpolation through non-linear data is inherently inaccurate. When you **STOP** an interpolated animation the current (static) state will be the one with the time closest to when you stopped the animation.

- Interpolated animations can be a bit slower than those at explicit states.

  Because you might choose a very large number of interpolated states D3PLOT does not store interpolated data for each frame, although interpolated coordinates are stored if space is available. Therefore there is a slight overhead as interpolated results are calculated "on the fly" during animation in these modes.

7.6.2.2. DISPLAY_MODES > the Display Mode Used for Graphics

DISPLAY_MODE The display mode used for graphics

D3PLOT supports four possible display modes, which affect animation performance:
DIRECT

Frames are calculated from scratch each time, no data being stored, so animation memory consumption is zero.

VECTOR

The data to redisplay each frame is stored in "vector" form in the D3PLOT (client) process itself. This is the default.

OBJECT

3D OpenGL only . Each image is stored as graphical "objects" in the OpenGL server, no animation data being stored in the client process.

Why so many different modes?

The answer is the trade-off of replay speed vs. memory use, and the need to optimise this in some cases. For a small model this is not an issue, but as you start to approach the limits of your computer with larger models you may find that you need to alter the animation method, or even move to using two computers in client/server mode. The following table summarises this:

Display mode	2D X-windows	3D OpenGL	Comments
DIRECT	Speed: Slow Mem: Zero	Speed: Slow Mem: Zero	Use if memory is short and you can tolerate slow speed.
VECTOR	Speed: Fast Mem: Small	Speed: Medium Mem: Medium	Best all-rounder, set as the programme's default.
OBJECT	*n/a*	Speed: Fast Mem: Huge	Again fast until you run out of memory, then dire. Runs well on remote server.

When do I need to change my display mode?

- If you get warnings about running out of memory during animation you may need to switch to **DIRECT** mode. Because of the way the operating systems on computers work it may be necessary to exit and restart D3PLOT to clear memory usage, then switch to this mode before rebuilding animations. Advice is given on-line about this if memory usage warnings are issued.

- If the default mode (**VECTOR**) is not fast enough, and enough memory is available, you can switch to **OBJECT** mode.

What is meant by "runs well on a remote server"?

Under OpenGL on LINUIX you are in fact using two processes:

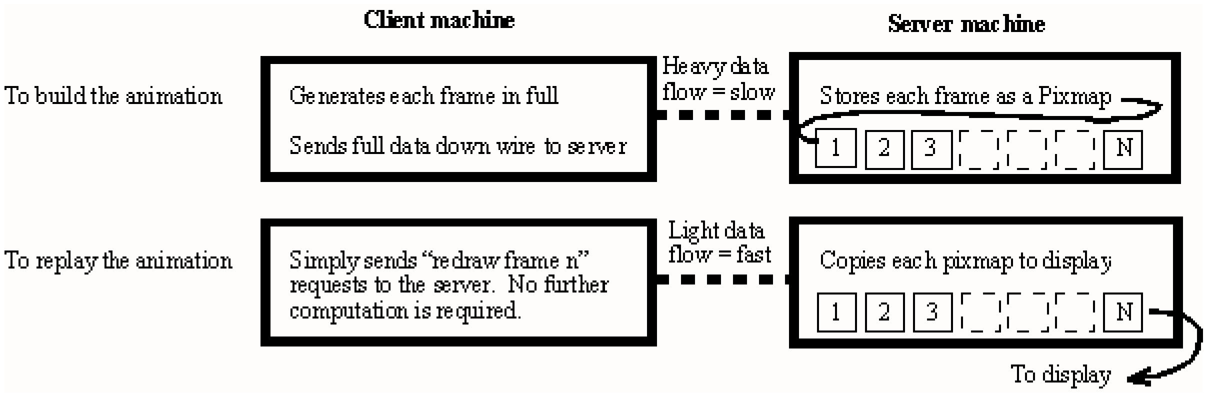
- The "client" is the D3PLOT session itself, computing what is to be drawn. It passes these drawing requests to ...

- ... the "server", which is the process on the machine responsible for turning drawing requests into raster images on the display.

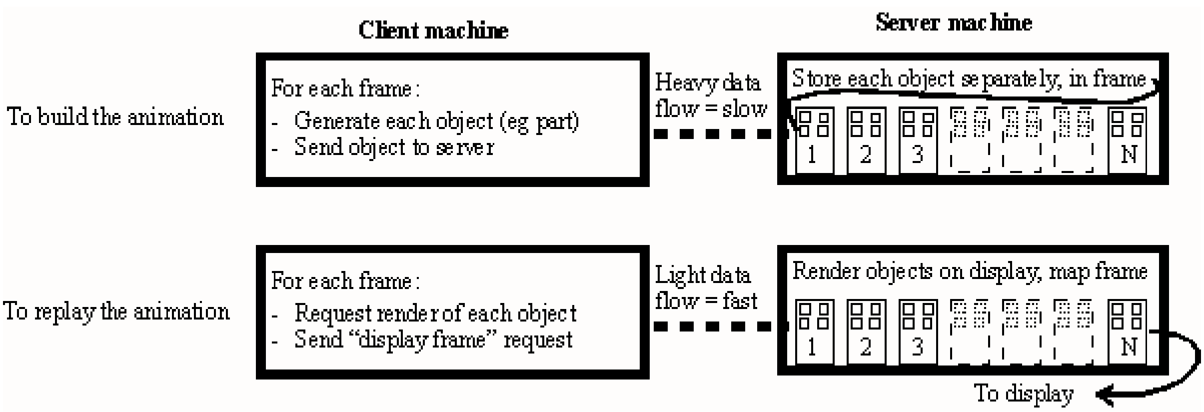
On a single machine the data transfer between the processes is fast - usually via shared memory - and OpenGL usage is generally direct from D3PLOT to the display card.

However there are some obvious advantages to separating the tasks between two machines: the client process can use all its host machine's resources to generate graphical requests, while the server can devote all its resources to displaying them. The disadvantage is that the data has to go down a piece of wire between the two machines, which may slow down data transfer. Therefore the best of all worlds is to store the graphics data in the server, whereupon the client only has to send a "draw this frame" request to render each frame of an animation.

OBJECT display modes achieve this end in related, but different, ways.



OBJECT mode:



Is no slower to compute in the client than **VECTOR** mode, but can take longer to draw during the first pass because of the overhead of building "objects" in the server.

Is reasonably fast to redraw (usually about 1.5 - 2x **VECTOR** speed), but this is a function of the number and complexity of the objects that make up the image.

Its memory requirements in the server are unquantifiable but large - typically up to 10 times that required for **VECTOR** mode.

Since it cannot tell how much memory the server is using D3PLOT is unable to protect you from the ill-effects of excessive memory consumption there. Ultimately it may lock up or crash if driven completely out of memory.

In 3D mode a viewing transform change imposes no speed penalty. All that is required is to send a new transformation matrix to the server.

If the contents of an object change then only that object has to be re-computed and re-sent to the server. So the "cost" of changes is proportional to the extent of the changes made.

Usually the memory consumption of **OBJECT** mode, and its less than phenomenal speed advantage over **VECTOR** mode, make its use impractical if both client and server processes are on the same machine. However if you have two machines available, or you are working from a remote host and want image transforms to be quick on your local machine, it is worth trying.

Setting up a remote client/server connection: (This is only possible on machines running X11, i.e. Linux or Unix hosts)

A description of how to display on a remote server is given in Starting the code , but briefly:

Prior to running the Shell to invoke D3PLOT:

On the client machine:	Set the **DISPLAY** environment variable to point to **<display>:0** on the server. (Eg **setenv DISPLAY server_name:0**)
On the server:	Make sure that windows can be opened by remote clients. (Under Unix the command is **xhost +**)

7.6.2.3. REPEAT > The Number of Times an Animation is Cycled Through

REPEAT > The number of times an animation is cycled through

By default an animation is repeated continuously until you stop it explicitly.

You can limit the number of passes through it to the options here, or to any number of your own choosing using the **Custom...** option.

If you limit the number of passes each **PLAY** request will result in that number of passes only. (The counter always starts afresh, it doesn't "remember" how many frames it displayed last time.)

7.6.2.4. DELAY > Delaying Playback Speed to Achieve an Explicit Number of Frames/Second

DELAY > Delaying playback speed to achieve an explicit number of frames/second

By default an animation is replayed at the fastest speed that the computer can manage.

Sometimes, especially in **PIXMAP** animation mode, this can be too fast and some frames get skipped. Alternatively if you are running multiple D3PLOT sessions, and you want animations to proceed simultaneously in several windows, you may find that you need to set an explicit display rate to stop one process "racing" ahead of the others.

Therefore it is possible to specify how many frames per second are displayed using the preset definitions here, or by using the **Custom...** option to select any frame rate.

Limitations of controlling playback speed

On most computers it will be difficult to achieve ***controlled*** frame rates faster than about 60 frames per second since 60Hz tends to be the resolution of the average computer clock, and finer timing is not achievable.

In addition the refresh rate of your display is significant. Most liquid crystal displays (LCDs) run at 60 frames per second (60Hz), and typical cathode ray tube monitors at between 60Hz and 100Hz. Attempting to animate at rates faster than this not worthwhile, and can be counter-productive.

For small models it is possible that D3PLOT will deliver the frames at a rate faster than the display refresh speed, in which case one of two things will happen.

1. If the graphics adapter has "wait for vertical refresh before swapping buffers" set then the animation rate will peak at the display's refresh speed, giving good results. Signs of this are smooth animation and the cpu usage of the D3PLOT process dropping as it waits before swapping buffers.

   This is the best outcome you can achieve and you need not take any action.

2. If the graphics adapter does not have "wait for vertical refresh" set the results can be "tearing" between successive frames as an image is redrawn part way through a buffer swap and is thus made up of data from more than one frame.

   If this occurs then you should turn on "Wait for vertical refresh" on your graphics adapter.

On Windows platforms this is usually achieved by ***< Right click on background >*, Settings, Advanced, *< graphics adapter name >*** and hunting through the options until you find the right setting.

On Unix / Linux platforms it is more difficult, and you may need to consult your hardware supplier for help.

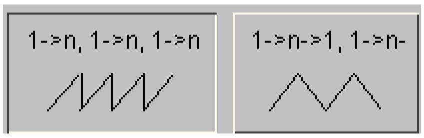
7.6.2.5. STYLE > Setting Playback to "Sawtooth" or "Modeshape" Styles

STYLE > Setting playback to sawtooth or modeshape styles

For transient analyses a "sawtooth" (1 => n, 1=>n) display generally looks more intuitive since it gives the impression of "start to finish".

However you can opt for "modeshape" (1 => n =>1 => n ...) mode if you wish (this is the default for static and eigenvalue analysis types).

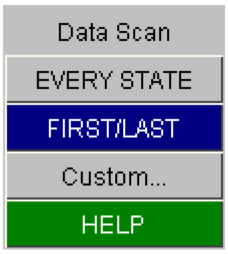
Examples of the two different modes (from their **Custom...** panel) are shown here in diagrammatic form.



7.6.2.6. DATA_SCAN > How States are Scanned to Find Max/Min "Automatic" Contour Levels During Animation

DATA_SCAN > How states are scanned to find max/min automatic contour levels during animation

When you perform an animation of data with contours in "automatic" mode D3PLOT has to scan through all candidate states to find their max and min data values so that it can set the contour levels.



If you have a lot of states this can be slow, and in many cases you will know that your data values rise (or fall) monotonically, and that using values from the first and last states only will bound all possible values in between. Therefore you have a choice of:

EVERY STATE This is the rigorous approach: every state is checked, and the true max and min values will be found. It can be slow if there are many states.

FIRST/LAST Only the first and last states selected for animation are scanned for their max and min values. This is the default.

In order to protect you from missing peaks and troughs if, during the assembly of an animation, a data value outside the expected max/min values is found while in **FIRST/LAST** mode you will be warned and offered the chance to swap back to **EVERY STATE** . However if you do swap back it will be necessary to rebuild any frames that have already been computed in order to make them have contour bands representing the new max/min values.

7.6.3. Improving Animation Performance

Improving Animation Performance

For small models this will not be a problem, but as the model size and number of states grows so you will see that animation performance degrades. This section describes how to speed up animations by reducing the load on your machine.

The key to fast animation is to reduce picture complexity, (simpler images have fewer vectors and so draw faster), and to reduce memory consumption (forcing your machine to page-fault with virtual memory usage will cripple its performance).

Choose an appropriate display mode

Clearly the time taken to draw each frame will increase in direct proportion to the number and complexity of the screen vectors used. Therefore you should aim only to display the minimum quantity necessary. The following table gives an approximate "cost" on a scale of 1 to 10 for the various display modes on OpenGL devices:

Display Mode		Relative Cost
LI	`(LINE)`	1
HI	`(HIDDEN-LINE)`	3
CT	`(CONTINUOUS_TONE)`	6
LC	`(LINE_CONTOURS)`	4
VEL/VEC	`(VECTOR/ARROW)`	5
SH	`(GREYSCALE SHADED)`	2 [(1)] or 4 [(2)]

SI	`(SHADED_IMAGE)`	3 (1) or 5 (2) or 10 (3)
ISO	`(ISO_SURFACE)`	6
CL	`(CLOUD)`	1

Notes: (1) No wireframe hidden-line overlay.

(2) With hidden-line overlay.

(3) With gouraud shading is turned off. (See SI)

Minimise contouring effort

Contouring can be especially graphics intensive and memory consuming:

- Turn the contour resolution to **Medium** or **Low** ;
- Use the minimum number of contour bands
- Turn off labelling of line contours in **LI** plots;
- Turn gouraud shading on for **SI** plots under OpenGL.

Reduce extraneous screen vectors

Turn off any extra information that is not definitely needed:

- Element and node labels;
- Node symbols;
- Element local triads.

Consider simplifying the display of entities. For example each 3D spring spiral contains about 70 vectors:

- Use "line" symbols for springs and seat-belt elements;
- Turn off contact segment hatching (broken lines are slow to render);
- Use free-edge or no overlay on data plots

If you are using cut-sections consider instead blanking the "unseen" parts of the model.

7.6.4. Animating Static and Eigenvalue (Modal) Analyses

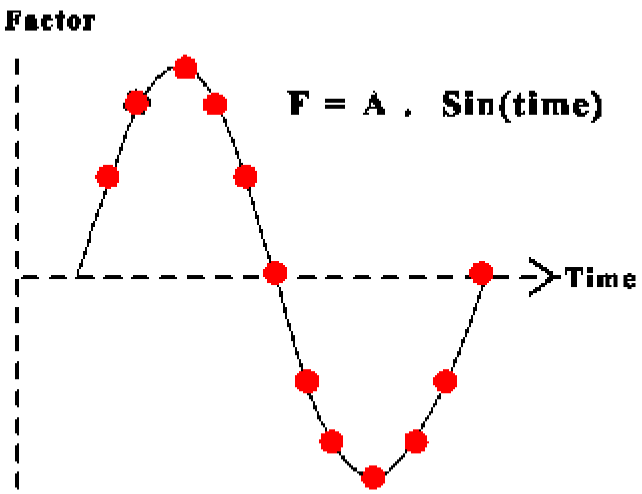
Animating Static and Eigenvalue (Modal) Analyses

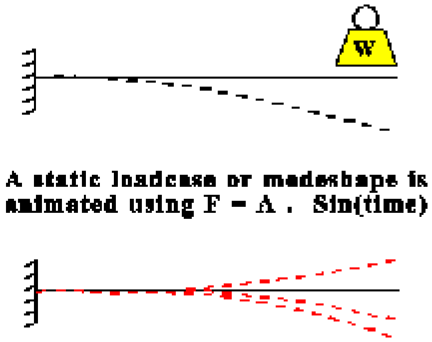
Recent versions of LS-DYNA incorporate the implicit solver, and this means that they can generate eigenvalue results. In addition it is possible to post-process static, eigenvalue and other solution sequences from Nastran analyses (see APPENDIX F).

Analyses of these types differ from conventional transient analyses in that each "state" is assumed to be:

- Eigenvalue analysis: A given modeshape
- Static analysis:The result of a given static loadcase combination.

Therefore when such analyses are animated it does not make sense to animate over states, rather a given "state" is cycled through a sinewave function to produce a "modeshape" plot.



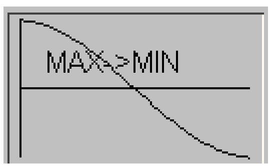


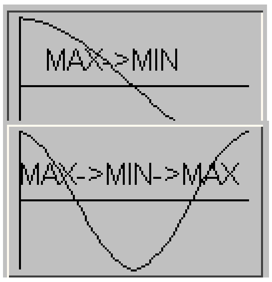
This has two implications for animation:

ANIM >, SET_FRAMES Setting the number of frames that are to occupy 360 of the sine wave

By default each 360 degree cycle of animation is split into 11 frames, which actually means 22 images, since the +ve and -ve cycles are symmetrical about their respective peak values.

The **SET_FRAMES** command in the **ANIM >** popup menu (which replaces the **SET_STATES** command in this context) allows you to choose a different number. More frames will give a smoother but slower animation.

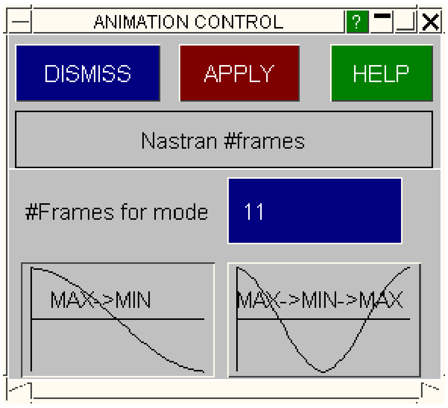




The **Custom...** option permits any number of frames to be defined, and also defines the period for the sine wave.

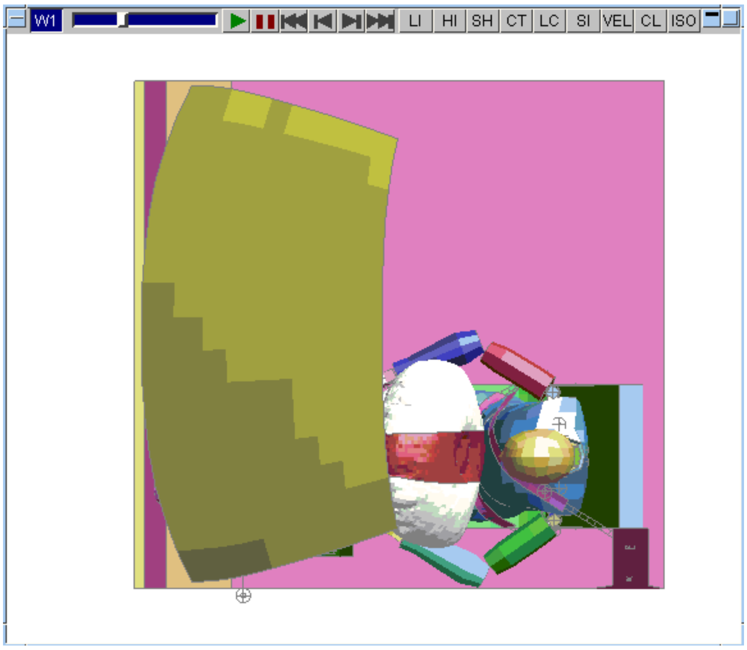
Normally the **MAX->MIN** option will be used, as this reflects the states internally to generate a 360 ^{0} animation from 180 ^{0} of frames.

The **MAX->MIN->MAX** option is only required when generating files for an external viewer that is not capable of "reflecting" a 180 ^{0} sequence into a 360 ^{0} one. It looks stupid on the screen, but will duplicate the frames to produce a full 360 ^{0} sequence in the file.



The frames slider cycles through the 0 - 360 cycle of frames, not through states

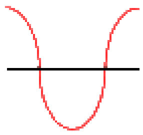
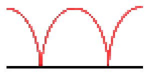
The frames slider, and other controls, cycle through the modeshape phase angle, not states. (See Basic animation controls)



Factors on results when animated by "modeshape"

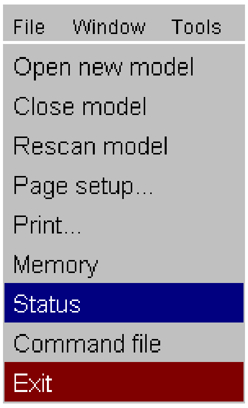
It is intuitively obvious that the factors on displacement to produce modeshapes need to be both +ve and -ve [factor = cos(time)]. It is less obvious what the factors on the corresponding results should be: magnitude values (such as von Mises stress) need a +ve/+ve, whereas direct stress tensor components (such a X direct stress) should be +ve/-ve, and components such as thickness should not vary at all.

Thus factors on data components through the 0 - 360deg vary as follows:

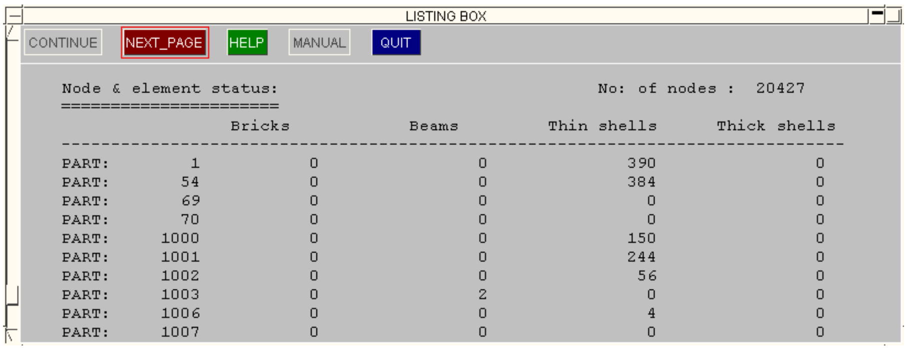
+ve/-ve factors, f = cos(theta) 	+ve/+ve factors, f = |cos(theta)  |	Unity factors, f = 1.0
[Sx,Sy,Sz,Txy,Tyz,Tzx] stress tensor [Ex,Ey,Ez,Exy,Eyz,Ezx] strain tensor Shell force & moment resultants [<outer fibre>] derived stresses [X,Y,Z] displacements	Everything not in the other two columns.	Thickness Shell Area Volume Outward normal Basic [X,Y,Z] coordinates Current [X,Y,Z] coordinates

7.7. STATUS Listing Programme Status

STATUS Listing Programme Status



The **STATUS** command has no sub-menus or arguments, it simply lists the current programme status. The following figure shows a typical listing:



		Bricks	Beams	Thin shells	Thick shells
PART:	1	0	0	390	0
PART:	54	0	0	384	0
PART:	69	0	0	0	0
PART:	70	0	0	0	0
PART:	1000	0	0	150	0
PART:	1001	0	0	244	0
PART:	1002	0	0	56	0
PART:	1003	0	2	0	0
PART:	1006	0	0	4	0
PART:	1007	0	0	0	0

The listing gives a cross-reference of the number of solid, beam, shell and thick-shell elements in each material; a summary of other entity types; a listing of each contact-surface, and the overall model dimensions.

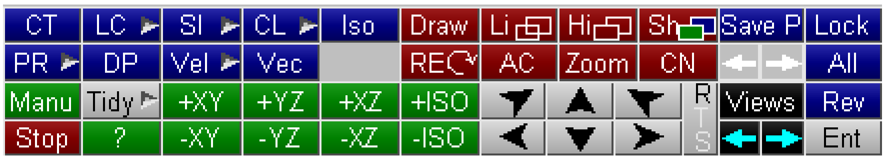
Lower down are also some key programme settings: blanking and volume-clipping switch status; the current in-core state number and time, etc. These are useful when D3PLOT is operating in command-line mode.

8. Viewing Control

VIEWING CONTROL

Controlling all aspects of viewing in the "Viewing Control" box

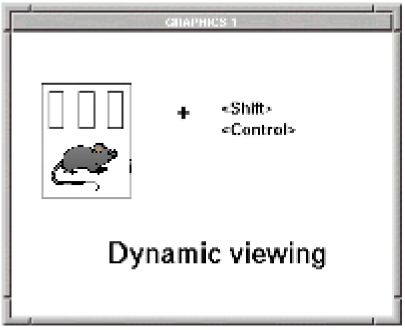
"Viewing" refers to the manipulation and presentation of images, rather than their actual generation. All viewing commands live in the "Viewing Control" box, located at the bottom right hand corner of the screen, and this section describes their use.



8.1. Dynamic Viewing (Using the Mouse to Change Views)

Dynamic Viewing (Using the Mouse to Change Views)

"Dynamic" viewing is the name given to the process in which you perform viewing transformations by moving the mouse around the screen. This is the most useful way of controlling views.



8.1.1. Graphics Modes During Dynamic Viewing

Graphics Modes During Dynamic Viewing

All dynamic viewing operations require a combination of two screen "meta" keys, (**<left control>** and **<left shift>**), and mouse buttons. The meta key(s) used dictates the graphics mode in which the image is transformed as follows:

<left shift> + **<mouse>**	Transforms the image in the current graphics mode. For example if it is a hidden-line plot, then dynamic viewing will take place in hidden-line mode.

`<left control>`	+ `<mouse>`	Transforms the image in "wire-frame" mode for the duration of the drawing operation. (ie no hidden-surface removal, or contours or lighting.)
`<left shift> &}<left control>}`	+ `<mouse>`	Transforms the image in pre-computed free-edge mode for the duration of the drawing operation. (ie wire-frame of free edges only, no hidden-surface removal, contouring or lighting.)

In the latter two cases the original drawing mode is always returned to at the end of the dynamic viewing operation. The wire-frame and free edge modes are provided to make transformations quicker for large models and/or slow computers: free edge is very fast.

For the last case, with **<left shift> & <left control>** held down together, the order of pressing and releasing the meta-keys matters: press **<left shift>** before **<left control>** , and release in the opposite order, otherwise you will (correctly) get the image redrawn in wire-frame mode as the **<left control>** key is pressed and released.

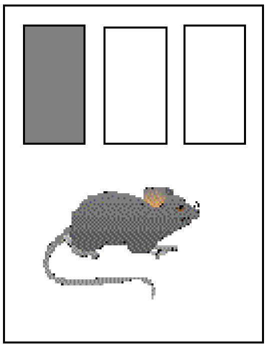
8.1.2. Dynamic Rotation

Dynamic Rotation

Dynamic rotation uses **<left mouse> + <left shift> &/or <left control>**

(The distinction between the keyboard meta-keys is explained in Graphics modes during dynamic viewing above.)

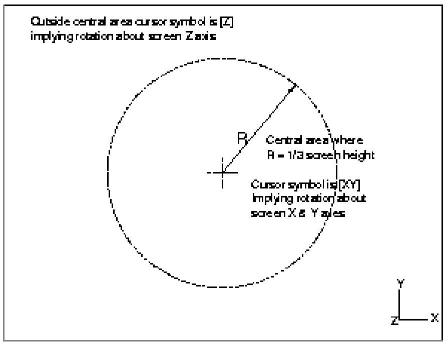
Rotation always take place in the **screen** coordinate system, and may be about the XY axes or Z: this depends upon the starting position of the mouse. This is shown in the next figure:



If the mouse initial position is **inside** the central circle (`radius (screen height/3)`) then rotation is about screen XY axes.

If the initial position is **outside** this circle then rotation will be about screen Z.

You can tell which mode you are in by the cursor symbol. This is **red** , and:



XY rotation uses	`[XY]`
Z rotation uses:	`[Z]`

The relationship between mouse and image motion is intuitive in both modes. It is as if you had grabbed a point on the object near you, (this side of the object centre plane), and used this to move the image about its centre:

XY mode	Moving the mouse left/right rotates about the screen Y axis; Moving the mouse up/down rotates about the screen X axis.
Z mode	Moving the mouse in a circular direction rotates about the screen Z axis.

Rotation remains locked in its initial XY or Z mode for the duration of a dynamic viewing operation, regardless of where you subsequently move the cursor to, until you release a mouse or keyboard button.

8.1.3. Dynamic Translation

Dynamic Translation

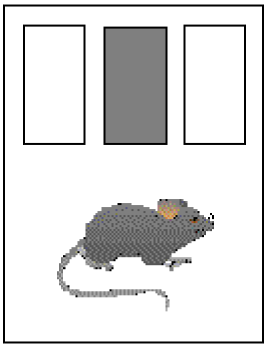
Dynamic translation uses **`<mid mouse> + <left shift> &/or <left control>`**

(The distinction between the keyboard meta-keys is explained in Graphics modes during dynamic viewing above.)

The cursor symbol is **yellow**, and looks like: 

Translation always take place in the **screen** coordinate system, in the X and Y directions.

The relationship between mouse and image motion is intuitive: the object tracks the mouse motion in the screen XY plane. The initial position of the mouse is irrelevant.

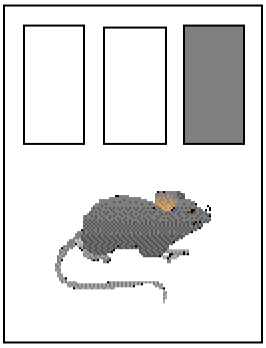


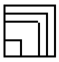
8.1.4. Dynamic Magnification (Scaling)

Dynamic Magnification (Scaling)

Dynamic scaling uses **`<right mouse> + <left shift>`**
`&/or <left control>`

(The distinction between the keyboard meta-keys is explained in Graphics modes during dynamic viewing above.)



The cursor symbol is **green**, and looks like: 

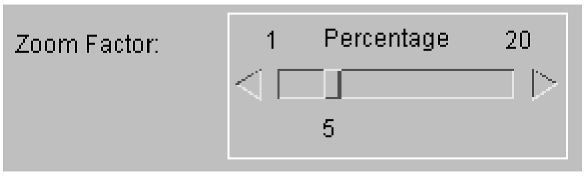
Mouse motion to the **right** and **up** makes the image larger, **left** and **down** smaller. The initial position of the mouse is irrelevant.

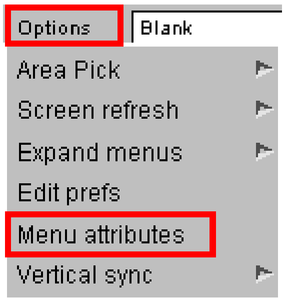
Dynamic magnification using the mouse scroll-wheel

If your mouse is equipped with a scroll-wheel then it will also perform dynamic magnification in the graphics window in which the cursor is present.

- Magnification is centred at the current cursor position unless " **CN** Centre node" has been used to lock centring on a node.
- Scrolling towards you magnifies the image scale.
- Scrolling away from you reduces the image scale.

By default each scroll wheel "click" will change the magnification factor by +/- 5%, but this can be changed using the **Options > Menu attributes** panel, and altering the **Zoom factor.**





8.1.5. Dynamic Viewing During Animation

Dynamic Viewing During Animation

On 3D devices operating in 3D mode you can carry out dynamic viewing during animation in exactly the same way as in static drawing. There should be no appreciable difference to the animation speed since all that is changed is the image transformation matrix.

8.1.6. 3D Mouse

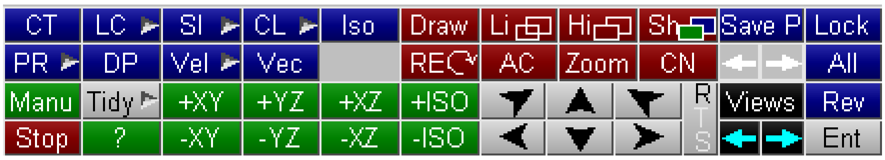
3D Mouse

From D3PLOT 11.0 onwards, dynamic viewing is also possible through the use of a 3D mouse. D3PLOT currently supports 3D mice produced by 3DConnexion. The 3D mouse is used in conjunction with a traditional mouse, by using one control to simultaneously pan, scale and rotate the model, while the traditional mouse is used for entity selection. Tilting or rotating the command cap of the 3D mouse will rotate the model around the geometric central point of the the visible entities. A rotation point can be manually set using " **CN** Centre node".

Different models of 3D mice also contain buttons that can be used within PRIMER for various operations. You can assign functions, macros and JavaScripts to the buttons on a 3D mouse by using the shortcut panel. See Shortcut Keys for more information.

8.2. Viewing Control Buttons

Viewing Control Buttons



+XY, +XZ, +ISO etc. — Pre-programmed view directions, also available from shortcut keys 1 through 8. For the starting view the default is +XY, this can be configured in the oa_pref file using:

```
d3plot*initial_view_orientation: < view >
```

ZOOM — Zooms in by using the cursor to pick a rectangular screen area that is to be enlarged to fill the screen - also available from shortcut Z.

CN — Picks a node about which dynamic rotation occurs. This remains active (with the **CN** button lit) until disabled by pressing **CN** again.

AC — Calculates the correct scale and centre position required to make the current image fit neatly onto the screen. This takes account of blanking, clipping, deformations, etc. Also available from shortcut button A.

RE — Forces a graphics refresh. This is occasionally needed if D3PLOT doesn't automatically update the display of data that is out of date.

Save P — Saves the current viewing attributes as a "Saved property" (see Saved Properties).

<= and => — Toggles backwards and forwards through any previously saved properties.

Views — Access to View manager (see VIEW MANAGER... Storing and retrieving "view" information)

<= and => — D3PLOT maintains a "history" of the last 100 views. The " **<=** " button toggles backwards through these and the " **=>** " one forwards through them.

Lock — Prevent currently blanked entities being unblanked.

All — Unblank all entities not locked.

Rev — Reverse blanking - also available from shortcut R.

Ent — Access to the entity panel (see ENTITY Switching the Display of Entity Categories On/Off) - also available from shortcut E.

Command-line commands are also available (e.g.):

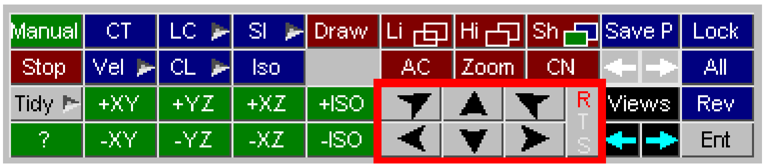
RM 30 0 0 - rotate (about model x,y,z axes) 30 degrees about the X-axis.
RS 30 0 0 - rotate (about screen x,y,z axes) 30 degrees about the X-axis.

8.2.1. Using the "Compass Rose"

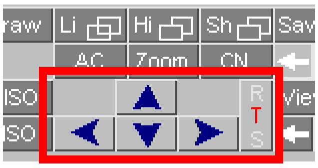
Using the "Compass Rose"

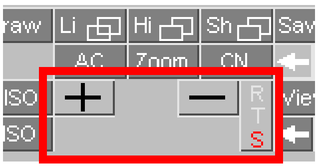
The "Compass Rose" provides three sets of buttons that allow the model to be rotated, translated and scaled with single mouse clicks. The **R T S** button toggles between **R** otation, **T** ranslation and **S** cale as shown here.

Timed action of all of these is possible if buttons are held down, and the consequent repeated actions can be stored in command files making it possible to programme and record viewing sequences. Use the Type drop-down menu to switch between rotation, translation and magnification options.



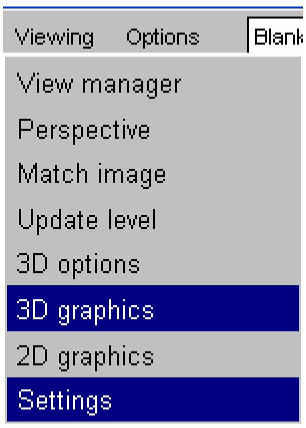
Default Rotation mode

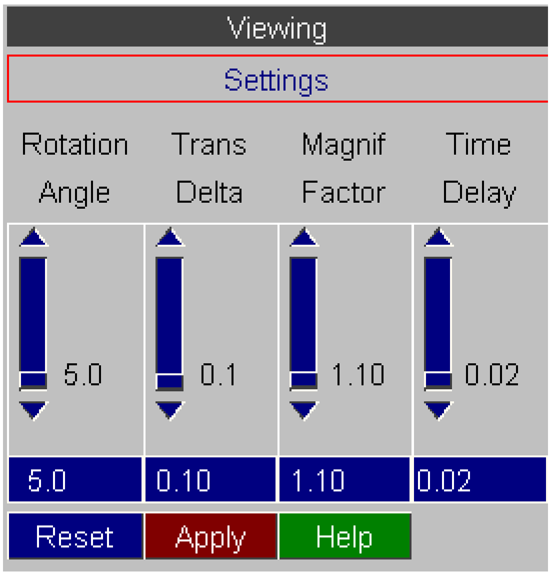

Translation mode


Scaling mode

Setting compass rose attributes

This panel allows control of the magnitude of transformation per click, and the time delay bewtween frames when a button is held down.





Programming transformations using the compass rose

The point has already been made that transformations using the compass rose are stored in command files just like any other command. It is worth repeating that "continuous" operations (those with a button repeating when held down) are also processed in this way: each repeat is stored like a separate "click" of the button. Thus a command file created in this way will have a sequence of many (possibly hundreds) of commands that are identical representing a series of repeated button presses.

You can use this to your advantage to make command files that rotate and/or translate the model automatically. For example you could pre-programme a "walk-through" of a structure in this way by saving the commands necessary to move your view point through and around and the model.

8.3. Options Under Viewing Menu

Options under Viewing menu

8.3.1. VIEW MANAGER... Storing and Retrieving "View" Information

VIEW MANAGER... Storing and Retrieving "View" Information

What is a view?

A "view" is all the information required to set up the current view of the object. In practice this means:

- The current rotation matrix (3 direction cosines).
- The current image centre location in space (x,y,z coordinate).
- The current magnification scale.
- The current perspective distance.

Up to 100 such views may be stored and retrieved at will from a file, and any number of such files may exist. The default file name is plot.view. A view is given a name and number when it is stored, and these are used when retrieving it. View files are binary compatible across platforms of the same word length (eg 32 bits), and are the same as those used by PRIMER.

Up to and including D3PLOT 9.3: Views are stored parametrically.

What this means is that views are not tied to a particular model, they will work for any model of similar dimensions. So if you are working on a set of variants of an analysis you can share the views on file between them: this is why they are stored in a separate, model-independent file. It is only when the shape and/or size of a model differs wildly from the original from which the view was created that this shareability fails.

From D3PLOT 9.3.1 onwards: Views are stored explicitly

The parametric method described above was not a success, as users wishing to compare models visually found it misleading. Therefore from D3PLOT 9.3.1 onwards views are now stored explicitly, and no account is taken of model size or position. Put qualitatively: the camera now stays in the same place with the same settings.

Retrieval of views is backwards-compatible. A view stored prior to D3PLOT 9.3.1 will read successfully into D3PLOT 9.3.1 onwards, but will be converted to "explicit" format if subsequently saved.

Using views

D3PLOT always has a current "view" definition. This dictates how the image will appear when a drawing command is issued. You can save the current view to file at any time. Likewise you can retrieve a stored view to replace the current one at any time.

The current view only exists in memory, and changing it has no influence on any views stored on file. (Indeed you don't need to have a stored view file: the default is none.)

Commands

STORE	Stores current view both in memory and in a view file. Click on a green (unused) view and type a name. Up to 100 views can be stored in a file, and views can be overwritten at will. If no explicit file has been opened the default file `plot.view` is opened automatically and used.
GET	Retrive from memory an existing view.
RENAME	Rename a stored view
DELETE	You can delete any existing vies
LIST	You can list information about stored views to screen
FILE	Define a file name in which views are to be stored

8.3.2. PERSPECTIVE... Setting Perspective Attributes

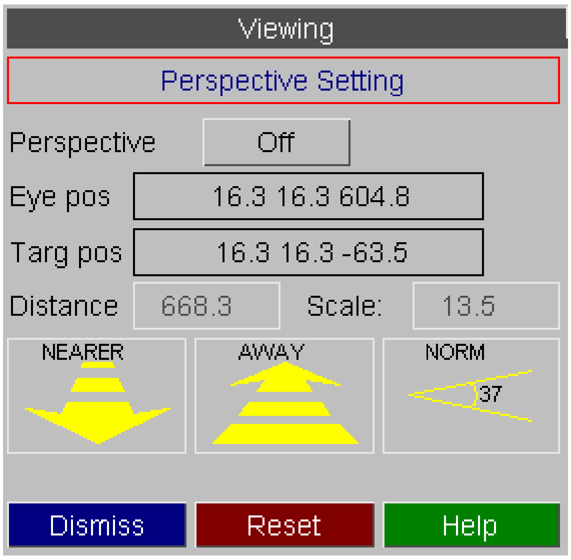
PERSPECTIVE... Setting Perspective Attributes

Use this option to switch on and adjust perspective settings.

D3PLOT will calculate the "bounding box" round your model and derive a default perspective distance of three times that value, which give a typical viewing angle of around 37 degrees.

Use **NEARER** and **AWAY** to adjust this, or type in a new **Distance** value.

If you get extremely close to the structure you may find that the overlay of hidden and shaded plots starts to come away from the underlying elements. This is a limitation of Z-buffered hidden-surface removal and a solution is given in controlling overlay quality in 3D mode.



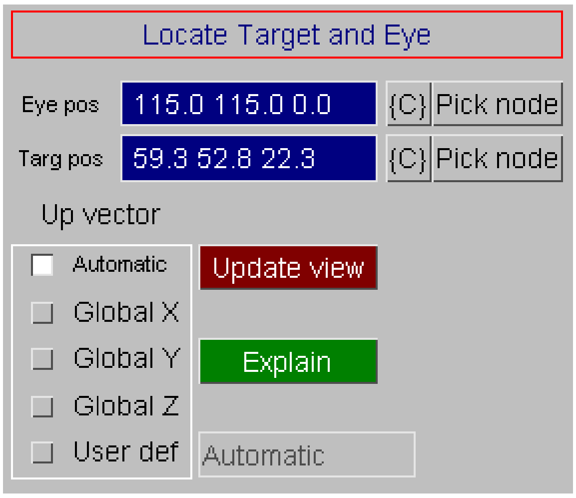
8.3.3. Locate Target and Eye

Locate Target and Eye

Normal D3PLOT viewing effectively positions the model in front of a stationary camera, then rotates, pans and enlarges it to place the desired region in the field of view of the lens.

However it is possible to set the "eye" (camera) position and also the "target" point on the structure at which the camera is pointing, and D3PLOT will compute the viewing transformation required to give the image from this point.

There are three components in a "Locate target and eye" definition:

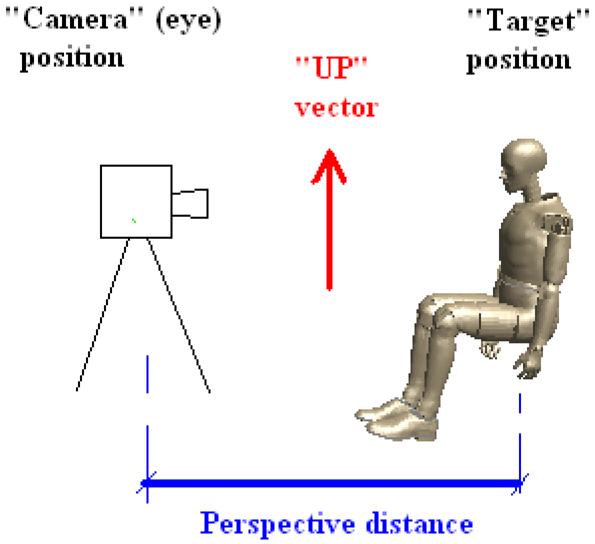


Target position	This is the coordinate in space at which the camera is pointing.
Eye position	This is the coordinate in space at which the camera (eye) is located.
"Up" vector	This is the vector defining "which way is up". Panning the camera up and down would move it up and down this axis.

The distance between the camera (eye) and target points is implicitly the current perspective distance, and this is reset when you **Update** the view. Perspective is switched on automatically if this is not already the case.

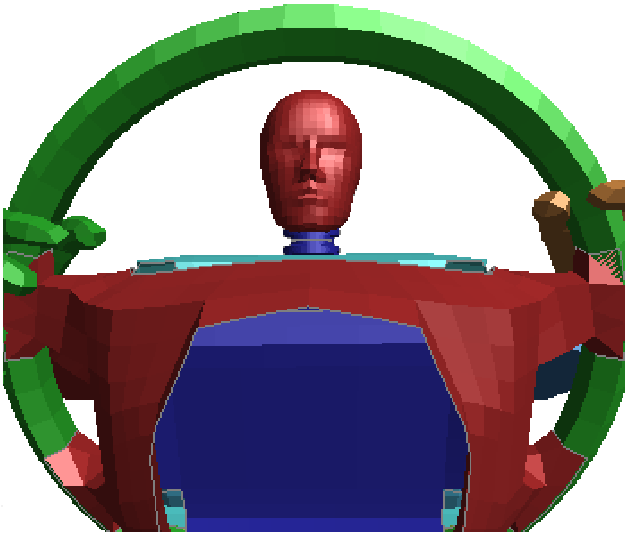
Both target and eye positions may be defined explicitly as coordinates in space, or you may screen-pick a node and its coordinate will be extracted.

By default D3PLOT tries to deduce the "Up" vector automatically, but you can override this by choosing a global vector, or by defining your own arbitrary vector.

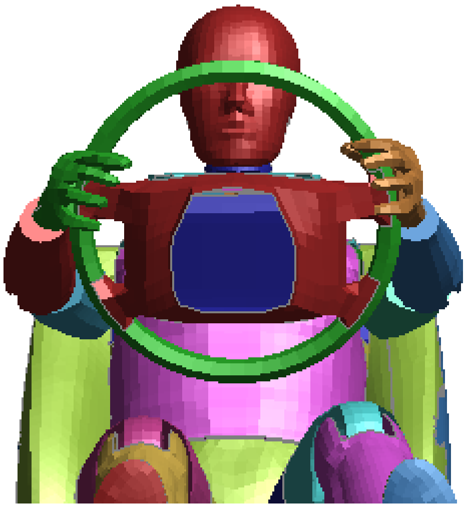


The relationship between Perspective Distance and Scale

If you use the "locate target and eye" feature you will almost certainly position your eye fairly close to the structure, which will bring you much closer than the normal perspective distance set by D3PLOT which is 3x the diagonal of the bounding box around the model. When the perspective distance becomes small the fore-shortening effect it causes becomes much more obvious.



In this image the target point is the dummy's nose, and eye point has been placed on the steering column just behind the wheel.



In this image the target point is the same, but the perspective distance has been increased by a factor of three, effectively moving the eye point backwards out of the paper.

Photographers will recognise that the perspective distance is, quite literally, the distance between subject and camera, whereas the scale is the "zoom power" (or, more precisely, focal length) of the lens on the camera. Both images above show the dummy head at approximately the same ***scale*** , but the difference in ***perspective distance*** gives rise to very different images.

If you are attempting to select viewing attributes to match an existing image you may find this quite difficult to achieve by hand since there are 11 independent variables to match in such an operation:

- Camera position (x,y,z coordinate = 3 variables)
- Vector from camera to subject (vector = 3 variables)
- "Up" vector (vector = 3 variables)
- Scale (1 variable)
- Perspective distance (1 variable)

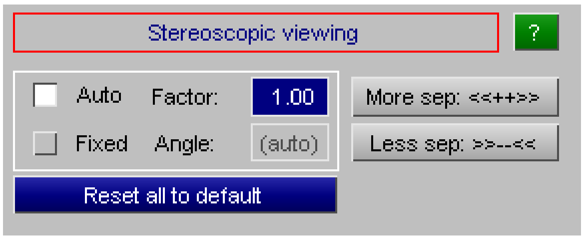
The **Match Image** function below will calculate this for you when given at least four points on the image and structure to match.

8.3.4. Stereo

Stereo

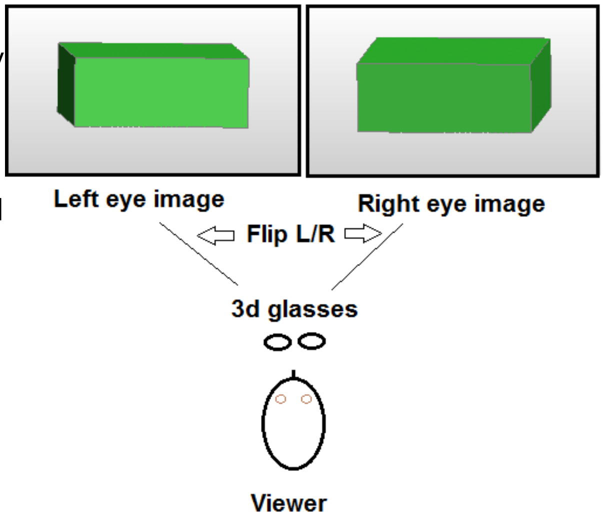
When "stereo" has been selected as the display device then all graphics windows will show stereo images which will give the illusion of 3D depth.

(This will only be available if the hardware supports stereo viewing and a stereo device has been selected for display, that is command-line argument "**-d=stereo**" which implies stereo OpenGL)



How stereo viewing works

The illusion of 3D depth is performed by showing slightly different images to left and right eyes.

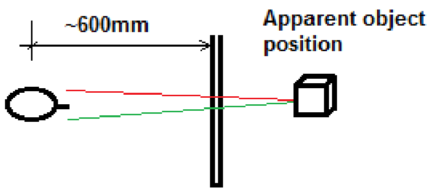


The hardware makes available "left" and "right" buffers, and swaps between them alternately - usually at 120 frames per second (120Hz).

Stereo glasses, synchronised with the display, shutter each eye in turn so that it only sees "its" image 60 times per second.

The 60Hz per eye refresh rate is faster than the eye can perceive, therefore the brain is fooled into thinking that it is seeing a single image viewed from two slightly different eye positions, and assembles this into the illusion of depth.

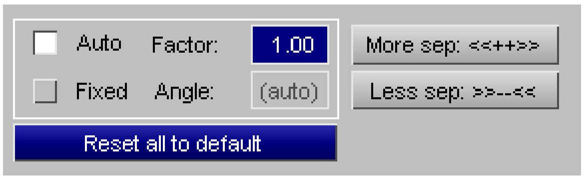
In order for this illusion to work the angular separation of the left and right images must approximate the "squint" angle that would actually occur when a viewer at a typical distance of about 600mm from a display looks at an object "behind" the plane of the display, ie somewhere "inside" the display.



Since eyes are typically about 80mm apart this means that each eye needs to "squint" inwards by an angle of approximately 2 degrees in order to give the correct apparent location of the object, and the left/right image projections are adjusted to give this angle.

Adjusting stereo separation

In practice users will not always be the expected distance from the screen, screen sizes vary, and indeed perception of depth varies from individual to individual. In other words there is no "perfect" setting that will please everybody all the time. In addition people with somewhat wonky vision (like the author!) may find too much stereo separation to be hard to view and - at worst - headache inducing.



Therefore D3PLOT has two modes for controlling stereo depth.

Automatic (default)	Sets a depth that is about 50% of the maximum stereo separation, ie depth effect, that an average user can tolerate. This is adjusted with image scale so that the depth effect remains constant regardless of scale.
Fixed angle	Allows you to set an explicit angular separation between the eyes, which remains fixed regardless of the image scale and perspective distance. This may be useful when using a stereo projector in a large room where the default geometry is far removed from that actually experienced by users. This will start at 2 degrees, and a value between 1 and 5 degrees is likely to be optimal.

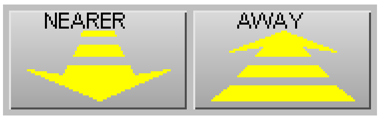
In both cases you can both increase and decrease stereo separation by using **More sep <<==>>** and **Less sep >>==<<,** which will adjust either mode by 5% per click.

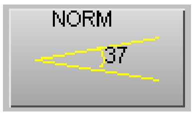
Reset all to default will restore the default setting.

Recommended practice

Achieving a satisfactory result is likely to be obtained by:

- Start with the default separation
- Try using the **Nearer** and **Away** buttons first to achieve the desired perspective effect
- If necessary apply more or less separation to adjust the effect.





... and ...

- If it all goes a bit pear-shaped (literally) use **Norm (37 deg)** to reset perspective and **Reset all to default** to reset stereo separation.

WARNING: Too much stereo separation can cause eye strain and/or headaches, so please don't overdo it!

Setting up stereo viewing

You usually only need to do this once, but getting it all to work that first time can be a challenge. This is only a brief description, if you need more help please contact Oasys Ltd with details of your equipment.

Hardware Prerequisites:

- The monitor or display device must be able to refresh at 120Hz.

  This is twice the normal 60Hz refresh rate of a standard LCD display, and typical monitors will not support this. Monitors that do are often described as being "3D capable", but the key metric is that 120Hz refresh rate. Sometimes such monitors are sold as having a "1ms response time", and most will actually refresh at up to 144Hz.

  The reason for this is that stereo requires separate "left eye" and "right eye" images to be drawn in quick succession, requiring twice as many redraws as a normal mono image. If the result is not to flicker this must be done at twice the normal 60Hz rate.

- The cable between graphics card and monitor must support this refresh rate.

  Direct "Display Port" or HDMI cables will usually support this, but simple passive adapters between these formats generally will not.
  If using DVI-D the cable must be a "dual link" cable with all pins present. It will be marked as such on the plugs.

Old-fashioned VGA cables almost certainly will not support it.

- You need some sort of 3D display hardware.

  This might be 3D glasses with an Infra Red (IR) or radio frequency emitter to sychronise with the graphics card, or some other hardware.

  In addition your graphics card must support stero rendering. Most modern cards will, but for some "engineering" cards such as NVidia's Quadro range it may be necessary to buy an additional adapter into which to plug the stereo hardware.

Software Configuration:

- The graphics driver software must support stereo viewing.

  This should not be a problem for any driver installed since about 2012, but it is worth checking that it does and whether any additional downloads are required.

- The stereo display must be the primary display on the computer.

  If you have multiple monitors and only one supports 120Hz refresh rates it must be the primary display.

- The stereo display must be configured to refresh at 120Hz or faster.

  Normally even a "3D capable" display will be configured by default to run at 60Hz, and you will need to configure it manually to run at the faster rate. On Windows 10 it is also necessary to designate the display as being in "3D display mode".

- It may be necessary to run a stereo setup "wizard" in the graphics driver configuration.

  This may perform first-time initialisations of hardware and software drivers. In the case of NVidia drivers it is also necessary to enable stereo in their "3D settings" list.

Avoiding, or at least mitigating,"slow flicker" effects caused by ambient lighting

You probably won't observe this, especially if the lighting in your environment is modern, high quality or natural. But if you experience an annoying "slow flicker" effect in your peripheral vision when using 3d glasses this section may help.

Stereo glasses alternate between left and right eyes at 120 frames per second, hence the requirement for a monitor running at a refresh rate of 120Hz, which means that each eye is "shuttered" at a rate of 60 frames per second (60Hz). Some forms of lighting, especially fluorescent tubes, can flicker a bit at the frequency of the electricity supply which tends to be 60Hz in the USA, but 50Hz in Europe and much of the rest of the world. Therefore when using stereo glasses it is possible for the "shutter" effect of the glasses to heterodyne with the "flicker" effect of the ambient lighting, giving a "slow flicker" effect in peripheral vision at the difference between these two frequencies.

In the USA where glasses and mains electricity supply are running at more or less the same frequency the effect is not likely to be noticeable, but when the difference is around 10Hz as in Europe and elsewhere the effect can be quite noticeable - and rather annoying. There are various things you can do to mitigate this:

- The cause is the ambient lighting in your environment, so if you can change this the problem will go away. The simplest solution is to turn it off, but replacing worn-out fluorescent tubes or moving to LED lighting may well solve the problem. Alternatively relocating the stereo hardware closer to a window, where natural light predominates (the sun doesn't flicker!) may help.
- If that is not possible then, depending on your hardware, you may not be forced to use exactly 120Hz as your screen refresh rate. Most 3d-capable monitors will support frequencies up to 144Hz, so if your graphics card will drive them at that frequency and - crucially - your 3d hardware will synchronise at different speeds, it is worth trying different values. Remember that the slow flicker is more or less at the ***difference*** between lighting and glasses frequencies, so changing that difference may move it to a regime to which the human eye is less sensitive.

Of the two solutions fixing the ambient lighting is much the best - possibly a "Health and Safety" issue?

8.3.5. Match Image

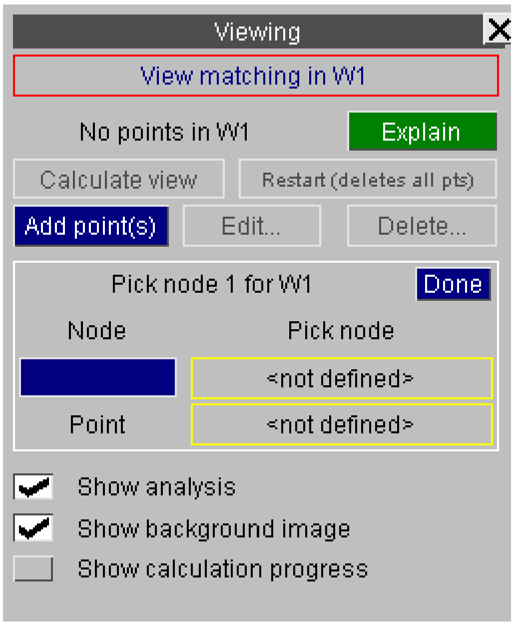
Match Image

Automatically aligns the current analysis image with the background by calculating the transformation parameters required.

Lining up an image requires the calculation of 11 unknowns:

- The camera position (3 coordinates)
- The direction in which the camera is pointing (3 vector terms)
- The "Up" axis of the camera (3 vector terms)
- The distance of the object from the camera, ie perspective distance (1 term)
- The focal length of the camera lens, ie image scale (1 term)

In the orthographic projection case, where the object is viewed in a parallel sided frustrum, the perspective distance can be omitted leaving only 10 values to be computed. However when photographs rather than computer-generated images are to be matched, which is normally the case, these implicitly use perspective projection and 11 variablkes must be computed.

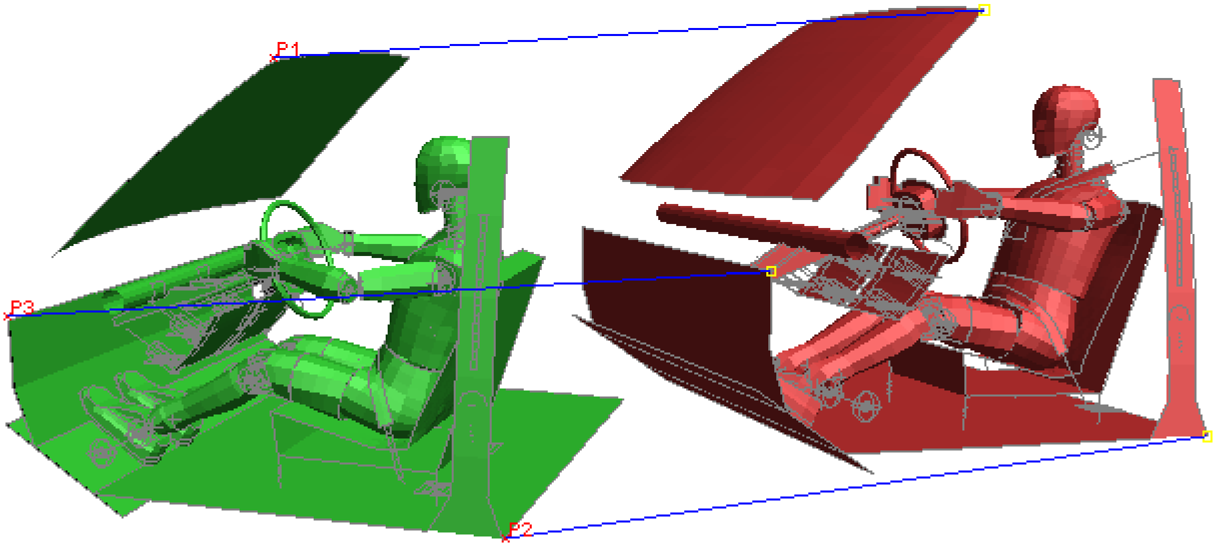


Each matching point on the image has a (x,y,z) coordinate, so to find 10 or 11 values a minimum of 4 points, giving 12 independent variables, are required. This calculation can be performed by D3PLOT if four or more well-chosen nodes on the model are matched to their corresponding points on the image, although in practice 5 or 6 points are required for a good match, mainly due to the difficulty of choosing well spaced points in the screen local Z (depth) direction.

Add point(s) Defining <node : point> pairs for matching

In the (artificial) example below the green image on the left has been read in as a background image, and the task is to get the red analysis image on the right to lie on top of it.

The user has defined 3 points so far: the nodes, identified by yellow pick symbols on the right, correspond to their matching points (red symbols and labels) on the left; the blue line shows which points and nodes are associated. These are screen-picked by selecting first the node, and then the corresponding point, and so on for the next pair.

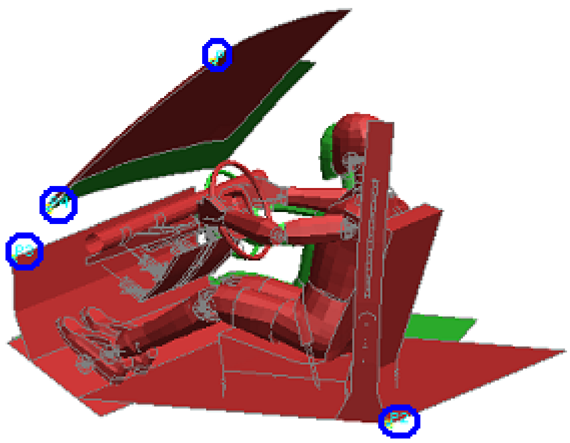


Calculate: aligning analysis with image

Once four or more <node : point> pairs have been defined it is possible to calculate the revised view. This will calculate the revised viewing parameters and update the image immediately. If the images can be matched and the points have been well chosen then the analysis should lie exactly over the target image.

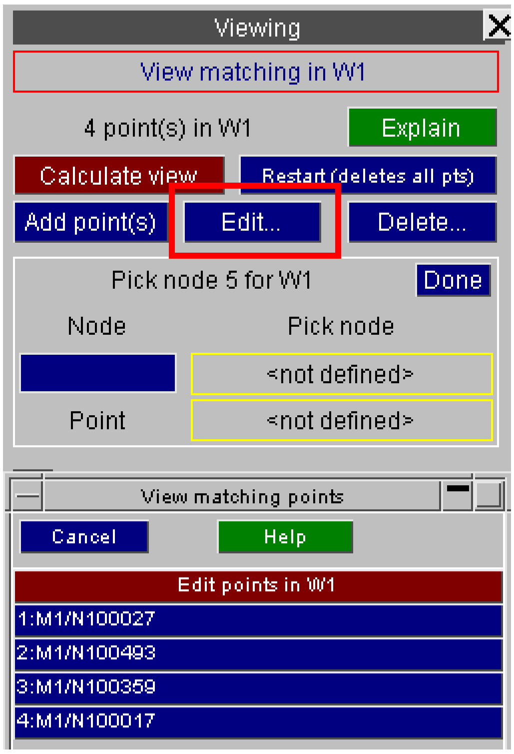
Edit...: correcting poorly chosen points

In the example below points have deliberately been chosen badly to obtain a poor match. (The error here is choosing points, ringed in blue, that lie more or less in a plane, making it difficult to calculate perspective distance correctly. In addition choosing only four pointscan be inadequate unless they are well spaced in all of (x,y,z) coordinate space, and more can be required for a good solution.)



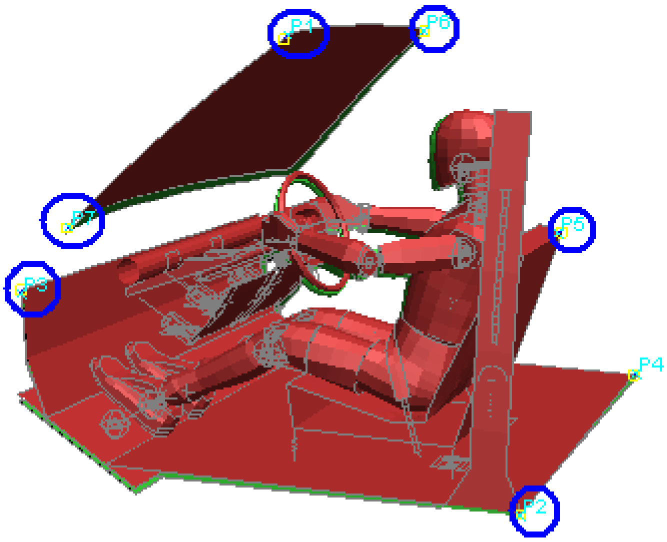
To edit a point screen-pick either its node or point (or select it from the menu), then repick its node or point.

Delete and **Restart**: Deleting points



Delete allows you to delete individual points by selecting them as above. Each point is deleted immediately.
Restart deletes all points letting you make a fresh start.

You can **Add** , **Edit** and **Delete** points in any order. Here is the example above with 6 points (circled in blue) chosen rather more judiciously, and it can be seen that the correspondence is now very good.



What is stored for matching

<Node : point> data is stored on a per-window basis, so it is not possible to apply matching data in Window #1 directly to windows #2, etc. However you can use the "Export view" function on the window's [--] options popup menu to export the current viewing parameters to all other active windows.

"Node" data is stored as a reference to a node in a model, and the current state's coordinate is used for matching purposes. Therefore if you need to match data during an animation you need to choose the state to be used for the matching process.

"Point" data is stored as a parametric (x,y) screen space coordinate, so points will remain valid so long as the aspect ratio of the window remains the same. However in most cases if a window is resized it is best to delete all the points and start again if further matching is required.

Trouble-shooting image matching

If you are having problems getting a good match between image and analysis the following trouble-shooting guide may help.

Choosing points that are all on a plane can cause problems

It is a common problem that many background images do not have much variation of depth - after all photographs are 2D - and as a consequence there is a tendency to pick points for matching that lie more or less on the same plane of depth with respect to the observer. This will usually give poor matching because it is very hard for D3PLOT to calculate perspective distance and scale when there is little variation of depth between points.

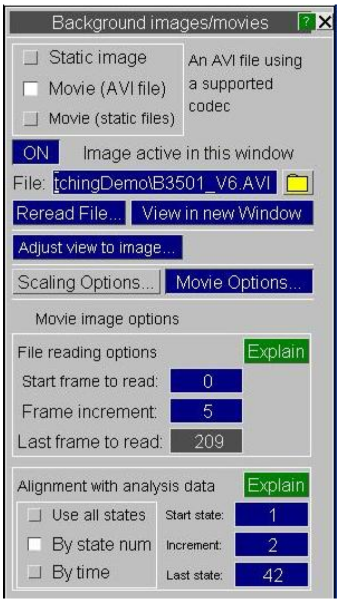
When selecting points the best match is achieved if you imagine a cube around the model, and try to pick points that are on a mixture of its near and far faces, as well as spread out left/right and top/bottom. There is no need to pick all 8 cube vertices, as four well-conditioned points are enough, but if perspective is active it is important to try to choose points that include a variation of depth.

Adding more points won't help if they are ill-conditioned

If the points you have chosen have not been defined accurately enough, or lie on a plane, then adding more similar points will not normally improve the solution - it will simply take longer to calculate the wrong answer.

It is far better to define 4 or 5 well-chosen points, and to delete any that only give a vague match between model and image.

Matching a model to a series of frames of an animation



At present image matching is "static". There is no provision for matching views separately to each frame of an animation. However, the model view can be matched to the first frame of the animation, which sets the model viewpoint at the real camera's position. Given that the camera's position is fixed relative to a known position throughout the animation, such as it moves with the model or is fixed to the ground, we can fix the model viewpoint, too in the same way.

In cases where the camera moves with the model you can use **Deform, Fixed Node** or **Shift Deformed** to track model movement.

In cases where the camera is fixed to the ground, you don't have to do anything because model viewpoint is fixed in the global coordinates at default.

After you have matched the model view to the first frame using the same technique as matching it to the background image, you need to match the timing, too. In this example we have a film with 0.002s per frame and a simulation analysis with 0.005s per state. To synchronize them we need every 5 frames of the film and every 2 states of the simulation analysis. You can set this at Movie Options.

8.3.6. UPDATE Level... Controlling the View Updating Frequency

UPDATE Level... Controlling the View Updating Frequency

D3PLOT has an **UPDATE_LEVEL** setting which dictates how often the view is updated following commands that change it.

1:NONE	The plot is never updated automatically. Changes only become apparent when you issue an explicit drawing command, eg **DR**, **CT**, etc.
2:MEDIUM (default)	The plot is updated immediately when any view control command is given, or any quick-pick command. The current image is amended as necessary following blanking, clipping, etc if any viewing command, **including dynamic viewing** , is used. In other words a viewing change command is tantamount to an explicit redraw command in the current mode which would, of course, reflect any changes in the model geometry.
3:FREQUENT	The plot is updated immediately as at level 2 above, but also following any menu-driven blanking, clipping, etc, command that would change the image if explicitly redrawn. Therefore the effects of blanking, etc are seen immediately.

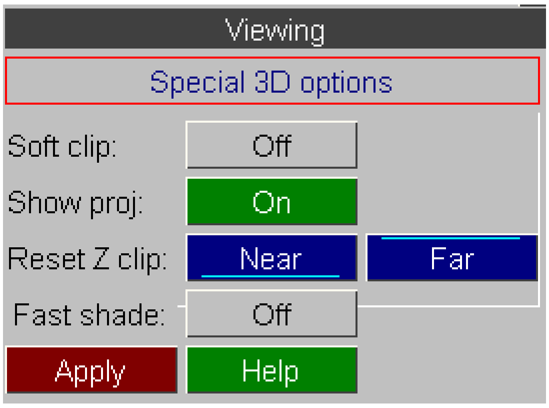
Note 1:	Level 3 is only recommended if you have a very fast display and/or a small model since it requires frequent redraws.
Note 2:	Users with slow devices and/or with large models may find that level 1 is preferable to decrease redrawing effort.

8.4. Special Graphics Options

8.4.1. 3D_OPTIONS... Further 3D Options

3D_OPTIONS... Further 3D Options

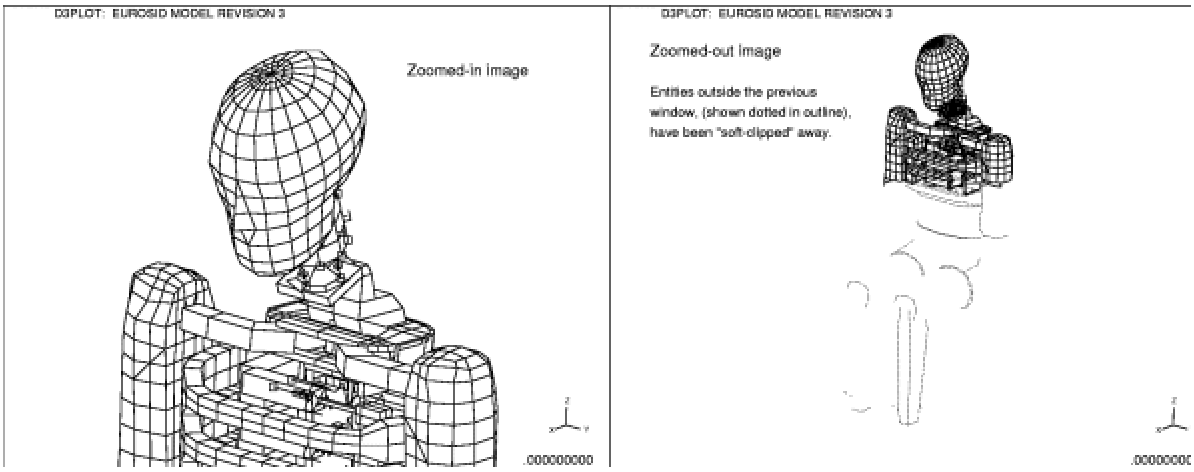
The **3D_OPTS...** button gives a control panel for further 3D options. These options are described below.



Soft clip Clipping graphics outside the current screen window

If you are dealing with a very large model, but are only looking at a small part of it, the 3D graphics driver can work unnecessarily slowly in its default mode of operation. This is because the whole model is sent to and manipulated by the graphics driver, despite the fact that you are only looking at a small part of it, in anticipation of your wanting to zoom out to see the whole of it.

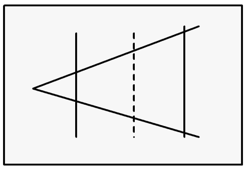
If you turn **Soft Clip** **on** , and redraw the image, the graphics will run faster. This is because the software has "clipped" (ie removed) those parts of the image not visible in the current window before sending it to the 3D graphics driver, so the 3D driver has to process fewer graphics entities. However this also means that if you zoom out those parts of the image outside the previous window will not be there. This is illustrated in the figures below.



In this example the user has zoomed in on the neck and upper chest region of a side-impact dummy (left hand image), and then zoomed out to what should show the full dummy. This exposes the jagged edges left by the 3D clipping algorithm.

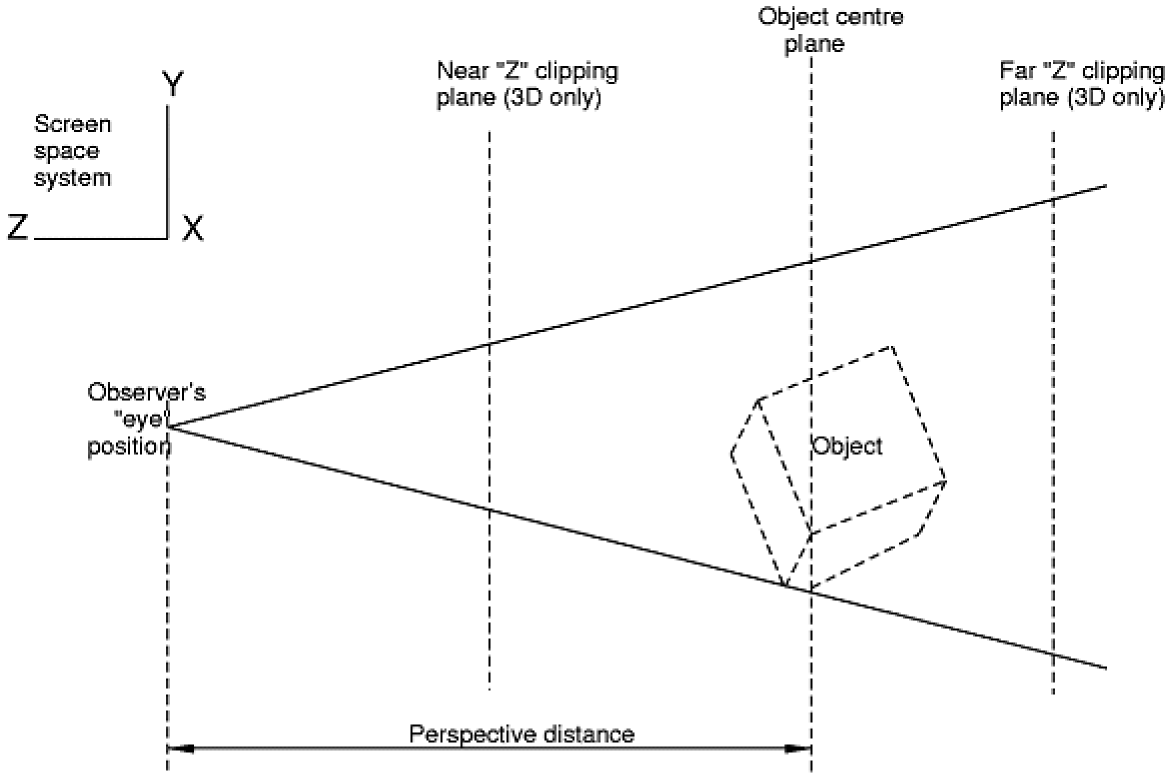
To see the missing elements you need to issue an explicit drawing command at the new scale to recalculate the clipping and send more elements to the 3D graphics driver.

SHOW_PROJ Showing the viewing frustrum



On 3D devices it is possible to show the current viewing "frustrum" at the bottom left corner of the plot by turning **SHOW_PROJ** on.

This shows the information in the figure below.



The frustrum shown here assumes perspective projection.

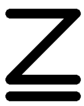
The Z clipping plane locations are shown when **SHOW_PROJ** is on, and this can be very helpful when using Z clipping, as otherwise it is easy to "lose" the clipping planes.

The default near and far plane positions are drawn in green, and the plane locations in blue. So you can visualise movement relative to initial locations.

Using the Z clipping planes

The Z clipping planes are shown in the figure above. There are two planes: a "near" and a "far" one, which the hardware uses to clip the image in the +/- screen Z axis.

By default they are set just outside the +/-Z limits of the structure (shown as green lines in the projection box), so that no clipping takes place, but when the 3D options box is mapped you can move them (shown as blue lines in the box) using the following mouse and keyboard meta-key combination:

`<right shift> + <left mouse>`	Moves the **near** clipping plane.	Cursor symbol is	
`<right shift> + <right mouse>`	Moves the **far** clipping plane.	Cursor symbol is	

`<right shift> + <mid mouse>`	Moves the **both** clipping planes.	Cursor symbol is	

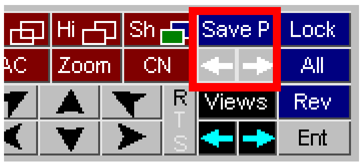
(Note that when the 3D options box is ***not*** mapped then the <right shift> and <right control> keys act exactly like their <left> equivalents, meaning that either side of the keyboard can be used for normal dynamic viewing.)

In all cases moving the mouse **up** moves the plane(s) **away** from you, and **down** moves **towards** you. This is a form of dynamic viewing: the planes move and the image gets updated as the cursor moves. It is recommended that you turn the **SHOW_PROJ** switch, described above, on as this will enable you to see the planes moving in the projection box.

To reset the planes to their default positions use the **Reset Z clip NEAR** and **FAR** buttons. This will reset them to their initial positions (shown by the blue lines in the projection box).

8.5. Saved Properties

Saved Properties



Saving and restoring the current view, colour, transparency and other attributes controlling the appearance of the image.

Saved properties were added in D3PLOT 11.0 and they perform the following functions:

- All the attributes controlling the appearance of the plot are recorded whenever a property is saved using **Save P**. The attributes stored are:
    - Colour, transparency, plotting mode, shading, overlays, explode vectors and blanking status of all items in the selected model
    - All settings in the Entity panel, ie visibility and labelling switches
    - The current view parameters: scale, orientation, position, perspective.
- Any number of properties can be saved in memory in D3PLOT, and you can scroll backwards and fowards through them using the **<=** and **=>** buttons.
- The attributes reset whenever a saved property is made current are controllable.
- Properties can be saved to file (extension .prp). This is an ASCII (human readable) file, written in a format that makes it portable between programmes, notably between D3PLOT and PRIMER, but others too if desired, making it possible to achieve the same image appearance in different programmes.

- Although the colour, transparency, plotting mode, shading, overlays, explode vectors and blanking status are stored with respect to the items in the source model, reuse of the properties file is not limited to this model and it can be used to set properties on any model that shares similar contents and label ranges.

There is some overlap of capabilities between the ability to toggle between and save "Views", and the ability to include the current view in a saved property. This is an historical accident due to the way the software has developed **.** Saved properties always contain view information and apply this by default when a property is restored. However this may not always be desirable.

8.5.1. Save P Saving the Current Attributes as a "Property"

Save P Saving the Current Attributes as a "Property"

Initially D3PLOT has no properties saved, so the saving button will show **Save P.**

Once you click on it to save a property it will be updated to be **SP i/j** where:

i is the current property number

j is the current total number of saved properties

You can still click on the renamed **SP i/j** to save further properties.

Cycling through saved properties using and 

Once you have saved one or more properties you can use the **<=** and **=>** buttons to cycle between them. Cycling left (**<=**) reduces the property number, and right (**=>**) increases it. It is possible to cycle backwards (left) to current property 0, which is explained below.

Property number 0, the "current" property

D3PLOT always maintains a "current" property which is what you see on the screen, and this is given the special number 0.

When you navigate to a saved property **i** this effectively copies that saved set of attributes to the current one, and likewise whenever you save a property you make a copy of the current property 0.

If you have navigated to a saved property **i** and you subsequently do something which changes the appearance of the image on the screen, for example blanking something, then the current property number gets reset to 0, and the **SP i/j** button will be updated to show **SP 0/j** .

This is because the current property no longer matches the saved property **i** , so it is no longer true to say that you are at property **i** . (The saved property **i** is not affected by this change: remember that making a saved property current copies it to current property 0, and it is only this current property that has been updated.)

Relationship between saved properties and the **Properties** panel

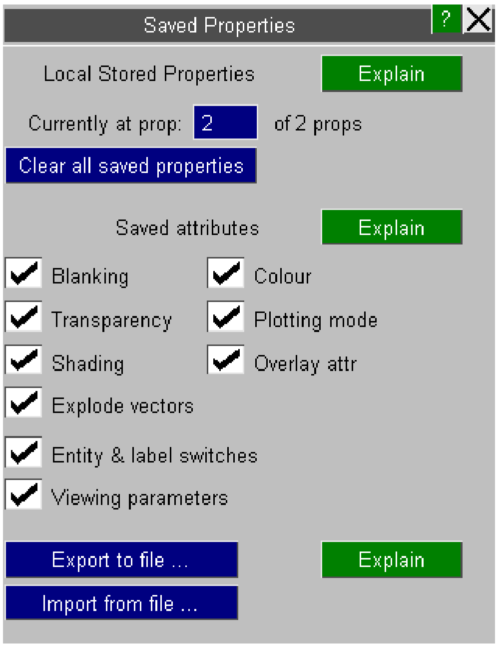
The **Properties** panel, described in Properties: Controlling colour, drawing style, transparency, lighting attributes and overlay of entities , gives more infromation about the current attributes of items in each model. It also allows detailed item properties to be viewed and changed. The save and reload functions in the Properties panel are the same as those here, and simply provide an alternative way of peforming the same tasks.

8.5.2. Options: Managing Saved Properties

Options: Managing Saved Properties

Hovering the cursor over the **Save P** (or **SP i/j**) button maps the `Save Props` popup in

which you can select **Options** to control saved properties.



This panel lists the current saved properties status, in this example currently at state 2 of 2, and allows you to select a saved property state directly by number.

Clear all saved properties deletes all saved properties in memory.

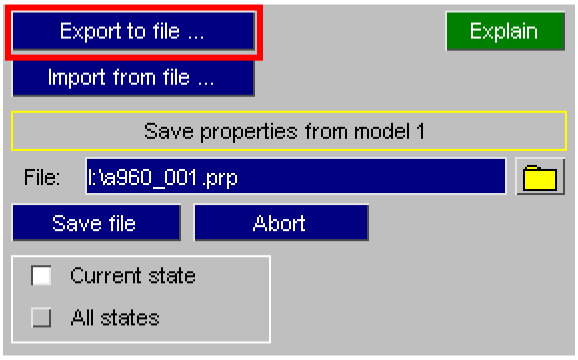
Saved Attributes lets you control which components of a property are updated when you navigate to a saved property. All attributes are always ***saved*** , this controls what is updated when the property is ***restored*** .

- **Blanking, Colour, Transparency, Plotting mode, Shading, Overlay** and **Explode vectors** are all attributes of the items in a model.

A saved property always contains ***all*** these attributes for all items in a model, regardless of whether or not they are currently visible. If items are added to the model after the property was saved their attributes will not be stored, since they weren't known about at the time of saving, so they will not be updated when the property is restored. (See below for further notes on the effects of changing model contents.)

- **Entity and Label switches** are the settings in the Entity panel and are model independent. A saved property contains the current status of all such switches for all possible item types, whether or not they are present in a given model.
- **Viewing parameters** are also model independent. The scale, orientation, location and perspective settings are stored.

Export to file... saving properties to file

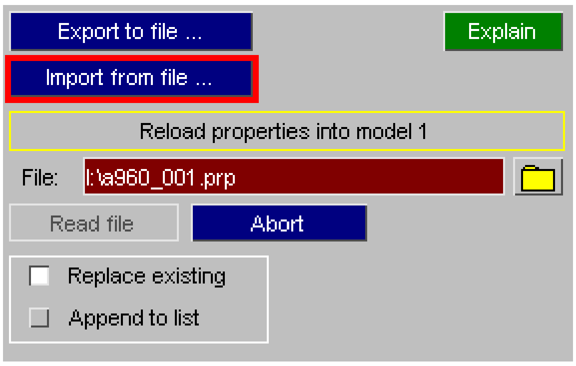


The complete contents of either the current property state 0 only, or all saved property states, can be saved to an external properties file.

This is an ASCII (human-readable) file that is designed to be both programme and model independent, and its format is given below.

The next free filename in the sequence **< *jobname* >_nnn.prp** will be presented as the default name, but you are free to use any name. Extension " **.prp** " is recommended though for compatibility with other Oasys Ltd LS-DYNA environment software.

Import from file... reloading properties from file



A previously saved file of properties can be reloaded into memory in this D3PLOT session, either replacing any existing properties or appending them to the current list.

The most recently created file in the sequence **< *jobname* >_nnn.prp** will be presented as the default filename and if, as here, no such file exists it will be listed on a red background and you will have to specify an alternative.

8.5.3. The Format of the Saved Properties (.prp) File

The Format of the Saved Properties (.prp) File

The saved properties file (.prp) is intended to be both programme-independent and model independent, so that attributes of a model's appearance can be shared between different programmes and variants of the same model. In particular the properties file can be shared between D3PLOT and PRIMER.

The file format is ASCII, so it is human readable and can be manipulated in a text editor, and its format is similar to LS-DYNA keyword format in that:

- Each data blocks begins with a "Keywords" that have an asterisk `*` in column 1.
- Any number of comment lines may be inserted, and they start with `$` , `%` or `#` in column 1.

However one significant difference is that all data is in free format, with no restrictions on field width or spacing between columns of data, so data will be formatted as:

string	a string of some number of characters
integer	either a decimal number, or a hexadecimal one if it starts "0x..."
float	a floating point number

A properties file contains the following blocks:

Header name	Status	Description	Notes
`*PROPERTIES`	Required	Defines the parameters of the following property state	This sequence of blocks is repeated for each saved property.
`*PROP_MASKS`	Required	Describes the format of the data to follow	
`*PROP_FAMILY`	*Optional*	Designates the family number of an adaptively meshed analysis. May be omitted for conventional analyses.	
`*PROP_DATA`	Required	Contains the actual property data for model items	

***PROP_SWITCHES**	*Optional*	Contains information about "entity" panel settings	
***PROP_VIEW**	*Optional*	Contains information about the current view settings	
***PROP_EXPLODE**	*Optional*	Contains information about the current explode part vectors	
***PROP_END**	Required	Acts as an "end of property definition" marker	
The following two blocks are written in D3PLOT property files only and are not strictly "property" data. They are provided for backwards compatibility with the older style of properties file used prior to D3PLOT release 11. If present these two sections only occur once, at the end of the file.			
***EXTERNAL_DATA**	*Optional*	Contains "external" data for "blob plots" as described in [External Data]	If present each of these sections appears once only.
***MODEL_TRANSFORM**	*Optional*	Contains "transformation" data as described in TRANSFORM	

Each block is described in more detail below.

For each saved property data blocks should appear in the following order:

```
*PROPERTIES
<code> <file version>
<saved id> <title>
```

<code>	string	is the programme name, here D3PLOT
<file version>	integer	is the version number of this file. This commences at 0 for release 11.
<saved id>	integer	is the saved property id, starting at 0 for "current".
<title>	string	is an optional title. At present this will be ignored.

This header block describes the basic parameters of the new saved property entry.

```
*PROP_MASKS
row 1: <keyword> <word><data mask>
row 2: <keyword><word> <data mask>
: :::
row n: <keyword><word> <data mask>
```

<keyword> <column> <mask>	string integer integer	One of a known series of mask names. The column number on the line, starting at 1. Integer or hexadecimal value giving bits used.

The purpose of this block is to allow different programmes, which will almost certainly store information in different formats, to stipulate how they are presenting data, and also to specify how many columns (words) of data will be supplied in the *PROP_DATA block below.

You don't need to understand this block unless you plan to generate property files yourself, or to read D3PLOT-generated property files into some other software. If this is the case please see " More about *PROP_MASKS " below.

```
*PROP_FAMILY
<family id>
```

<family id>	integer	The adaptively meshed family number, starting at 0, for which the following *PROP_DATA information applies.

This header can be ignored except in the case of adaptively remeshed families which repeat the

*PROP_FAMILY
<family id>

*PROP_DATA
<data for family>

sequence for each family in the analysis. This is becase each family can have different numbers of nodes and elements.

```
*PROP_DATA
row 1: <item type> <start label><end label><word #1><word #2>...<word #n>
row 2: <item type> <start label><end label><word #1><word #2>... <word #n>
: ::::::
row n: <item type> <start label><end label><word #1><word #2>...<word #n>
```

<item type> <start label> <end label> <word #1> <word #2> <word #n>	string integer integer integer integer integer	Item name, eg NODE, PART, etc The first label in the range, or FIRST or ALL The end label in the range, or LAST. Omitted if the start label is ALL. The first word of data, ie column 1 The second word of data, ie column 2 The last word of data, ie column n

The storage method here echoes the internal runlength-encoded format in which all items in the label range <start> ... <end> have the same property values.

ALL is used instead of <start> .. <end> labels when all items of the type share the same attributes.

FIRST is used in place of label <start> if this is the first item of its type, and **LAST** in place of label <end> if it is the last label. This is so that other models, perhaps with slightly different label ranges, will still apply the properties correctly.

Data words #1 to #n must be supplied for every item even if they do not contain any useful data, in which case they can be zero. The number of words expected on each line, #n, is inferred from the highest <column> entry in the preceding ***PROP_MASKS** block.

```
*PROP_SWITCHES
row 1: <item type> <drawn><labelled><named>
row 2:<item type> <drawn><labelled><named>
::: :
row n: <item type> <drawn><labelled><named>
```

<item type>	string	Item name, eg NODE, PART, etc
<drawn>	integer	Whether this item is drawn
<labelled>	integer	Whether this item is labelled
<named>	integer	Whether this item is named

This data block is optional: if omitted the "entity" panel settings will be left unchanged when the file is read.

Each data field <drawn>, <labelled>, <named> is, at its simplest, 1 for true and 0 for false. However within D3PLOT some item types have sub-keywords, and further bits can be used to denote the individual status of these.

```
*PROP_VIEW
Matrix row 1: <X cosine><Y cosine><Z cosine>
Matrix row 2: <X cosine><Y cosine><Z cosine>
Matrix row 3: <X cosine><Y cosine><Z cosine>
Offsets:<X trans><Y trans><Z trans>
Scale:<Scale factor>
Perspective:<On/off><Distance>
```

<X/Y/Z cosine>	float	The X/Y/Z components of the unit cosines for that matrix row
<X/Y/Z trans>	float	The X/Y/Z component of the translations required to position the model in front of the eye position.
<Scale>	float	The scale factor from model space to screen (4096 x 4096) space
<On/off>	integer	Whether perspective is on (1) or off (0)
<Distance>	float	The perspective distance (from eye position to model centre)

This data block is optional. If it is omitted the view will not be updated when the file is read.

```
*PROP_EXPLODE
Part ID <X component><Y component><Z component>
Part ID <X component><Y component><Z component>
Part ID <X component><Y component><Z component>
```

<Part ID>	Integer	The part ID for that row of the table
<X/Y/Z component	Float	The X/Y/Z component of the translations required to position the part in its exploded position

This data block is optional. If it is omitted the exploded part information will not be updated when the file is read.

```
*PROP_END
```
(This block has no data)

This block signifies the end of the current property definition.

8.5.3.1. Example Properties File

Example Properties File

Here is an example properties file from a small model.

```
$ File J:\sled_model_binout\new_lg09_008.prp written at Wed Dec 07 15:12:34
2011
$
$ D3PLOT Version  : 11
$ File   Version  :  6
$
$
*PROPERTIES
$
$ Code    File version
D3PLOT              6
$ State id       Title
         0
$
$
*PROP_MASKS
$
$ Attribute     Word       Bits
$ ----------------------------
BLANKED           1      0x2000
$
MODE_MASK         2         0x3
BRIGHT_MASK       2        0x3c
SHINE_MASK        2       0x3c0
OVLAY_MASK        2       0xc00
OVL_R_MASK        2      0x7000
```

```
OVL_G_MASK       2      0x38000
OVL_B_MASK       2     0x1c0000
OVL_CURRENT      2     0x200000
OVL_DEFAULT      2     0x400000
ENTITY_DEF       2     0x800000
$
ALPHA_MASK       3   0xff000000
RED_MASK         3         0xff
GREEN_MASK       3       0xff00
BLUE_MASK        3     0xff0000
$
$
*PROP_FAMILY
$ Family id
         0
$
$
*PROP_DATA
$


$ Type                                      Label #1    Label #2     Word #1
Word #2    Word #3
$ -------------------------------------------------------------------------
-----------------
$
NODE                                             ALL                       0
0x1ffeab 0xffffffff
BEAM                                             ALL                       0
0x3c7eab 0xffff00ff
SHELL                                          FIRST          64           0
0x207eab 0xff0000ff
SHELL                                             65         128           0
0x238eab 0xff00ff00
SHELL                                            129         256           0
0x3c0eab 0xffff0000
SHELL                                           9055        9056           0
0x307eab 0xff9900ff
SHELL                                           9057        9058           0
0x22feab 0xff00bbff
SHELL                                           9059        LAST           0
0x3f8eab 0xffffff00
SPRING                                         FIRST           1           0
0x23dd37 0xff00ffa8
SPRING                                             2           2           0
0x378d37 0xffa8ff00
SPRING                                             3           3           0
0x3d8d37 0xffff7f00
SPRING                                      10000093    10000096           0
0x3f8d37 0xffffff00
SPRING                                      10000097        LAST           0
0x3c7d37 0xffff00ff
SBELT                                 FIRST          107           0    0x23fd37
0xff00ffff
RETRACTOR                                          1           1           0
0x207d37 0xff0000ff
SLIPRING                                           1           2           0
0x238d37 0xff00ff00
PRETENSIONER                                       1        LAST           0
```

```
0x3c0d37 0xffff0000
JOINT                                        FIRST          14            0
0x207eab 0xff0000ff
JOINT                                           15        LAST            0
0x3c0eab 0xffff0000
GLOBAL                                         ALL                        0
0xd37 0xffcdcdcd
PART                                         FIRST           1            0
0x207eab 0xff0000ff
PART                                             2           2            0
0x238eab 0xff00ff00
PART                                             3           3            0
0x3c0eab 0xffff0000
PART                                             4           4            0
0x3f8eab 0xffffff00
PART                                             5           5            0
0x3c7eab 0xffff00ff
PART                                             6           6            0
0x23feab 0xff00ffff
PART                                             7           7            0
0x307eab 0xff9900ff
PART                                             8           8            0
0x22feab 0xff00bbff
PART                                             9           9            0
0x23deab 0xff00ffaa
PART                                          2001        LAST            0
0x21feab 0xff0077ff
$
$
*PROP_SWITCHES
$


$ Entity type switches       Drawn     Labels    Names
$ -----------------------------------------------------
$
NODE                           0          0          0
BEAM                         0x1          0          0
SHELL                        0x1          0          0
SPRING                       0x1          0          0
Belt_type                   0x1e          0          0
JOINT                          0          0          0
GLOBAL                         0          0          0
PART                           0          0          0
$
$
*PROP_VIEW
$
$ Current viewing attributes
$ --------------------------
$
Matrix row 1:    9.845316E-001    1.740706E-001    1.997507E-002
Matrix row 2:   -5.696383E-002    2.101849E-001    9.760019E-001
Matrix row 3:    1.656945E-001   -9.620420E-001    2.168492E-001
Offsets:        -3.505000E+002   -4.200000E+001    2.824991E+002
Scale:           2.318954E+000
Perspective:                 0    4.503000E+003
$
$
```

```
*PROP_END
$
$
$
$
*EXTERNAL_DATA
$
External data
0 0 1 1 20 20
20 0 0 0 1
0.000000e+000 0.000000e+000 1.000000e+000 1.000000e+000 1.000000e+000
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
$
$
*MODEL_TRANSFORM
$
0 0.000000e+000 0.000000e+000 0.000000e+000
0 0 0.000000e+000
0 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
0 0.000000e+000 0.000000e+000 0.000000e+000
$
$ End of file
```

8.5.3.2. More About the *PROP_MASKS Block

More About the *PROP_MASKS Block

You only need to understand property "masks" if you plan to create your own property files, or to read the D3PLOT-generated ones into 3rd party software.

A "mask" defines the bits in a word that are used to contain data. In this context a "word" is always a single precision 32 bit integer, so you will be defining which of these 32 bits contain the data you want.

As an example let us take the problem of defining colour, which is specified by 4 components, generally known as RGBA in computer graphics:

Component	Property mask	Description
R ed	`RED_MASK`	For each of red, green and blue the value must be in the range 0 to 100%
G reen	`GREEN_MASK`	
B lue	`BLUE_MASK`	

A lpha (transparency)	**ALPHA_MASK**	A value must lie in the range 0% (fully transparent) to 100% (fully opaque)

Therefore bright red, with no transparency, would comprise 100% Red, 0% Green, 0% Blue, 100% Alpha.

Example 1: External data contains each colour component as a separate floating point value in the range 0.0 to 100.0

In this case the easiest solution would be to express your colours as 4 separate values. These must be integers, and the full bit field must imply 100%, so the easiest solution would be to convert the floating point range 0.0 to 100.0 into values in the range 0 to 255 by multiplying by 2.55 and writing the result as integers. The data masks you define might then be:

RED_MASK1255 **GREEN_MASK2255** **BLUE_MASK3255** **ALPHA_MASK4255**	Each colour channel is defined in a separate integer word Red = word #1, Green = word #2, Blue = word #3, Alpha = word #4 and lies in the range 0 - 255

And a typical property line to define some shells with labels 1 to 10 that are cyan (green + blue) and 50% transparent would then be

Item name	Start label	End label	Word 1: Red value	W2: Green value	W3: Blue value	W4: Alpha value	.. further columns
SHELL	**1**	**10**	**0**	**255**	**255**	**128**	**. . .**

The choice of columns 1 to 4 for the RGBA components is arbitrary, you could choose any columns you like.

Example 2: External data contains each colour component packed in a single 32 bit word

A more compact, and very common, way of storing RGBA data is to express each colour component in the range 0 - 255, which requires 8 bits or 1 byte, and to pack these four bytes into a single 32 bit word. Drawn as a diagram we could express the 32 bits in this word as:

Highest byte: Alpha bits	Blue bits	Green bits	Lowest byte: red bits
AAAAAAAA	**BBBBBBBB**	**GGGGGGGG**	**RRRRRRRR**

We can now define our colour masks, assuming that the colour word is in column #1, as

```
RED_MASK10x000000ff
GREEN_MASK10x0000ff00
BLUE_MASK10x00ff0000
ALPHA_MASK 10xff000000
```

Hexadecimal (0x...) format has been used here, but the values could equally well - if less conveniently - be expressed in decimal. For example the Red mask `0x000000ff` is the same as decimal `255` , and it would be legal to use that instead. Using this format our 50% transparent cyan shells would now be defined more compactly as:

Item name	Start label	End label	Word 1: RGBA	.. further columns
`SHELL`	`1`	`10`	`0x80ffff00`	`...`

Again hexadecimal has been used here, since the decimal equivalent would be an unwieldy negative number.

What property masks are required?

You only have to provide property masks for the values you want to change. When property files are read in they only overwrite the attributes that they define so, for example, if you only included blanking information in a file the colour and lighting attributes of the model would be unchanged when it was read. Another example might be that you only have RGB colour information, and no Alpha (transparency) data. In that case omitting the Alpha mask and data word would leave item transparency unchanged when a file is read.

Which columns may data occupy?

Up to 20 columns of data may be provided, numbered 1 to 20, and any attribute may exist in any column. When the *PROP_MASKS data block is read the highest column number is remembered and the subsequent *PROP_DATA block must contain that many columns of data on each line. It doesn't matter if data in a given column is not read, for example if you already have formatted data and you want to ignore some of it simply define masks that only specify the data you want.

Valid property masks for D3PLOT:

Mask name	Meaning
`BLANKED`	The bit(s) used to designate that an item is blanked, ie blanked (non-zero) or unblanked (zero)
`MODE_MASK`	The display mode for element graphics: 0= wireframe, 1 = hidden, 2 = shaded, 3 = current

<table>
<tr><td>ALPHA_MASK</td><td>The Alpha (transparency) bits. 100% = opaque.</td><td rowspan="6">It is assumed that a fully occupied bit field is 100% of the given component value for all these types</td></tr>
<tr><td>RED_MASK</td><td>The Red bits</td></tr>
<tr><td>GREEN_MASK</td><td>The Green bits</td></tr>
<tr><td>BLUE_MASK</td><td>The Blue bits</td></tr>
<tr><td>BRIGHT_MASK</td><td>The diffuse brightness</td></tr>
<tr><td>SHINE_MASK</td><td>The specular brightness (shininess)</td></tr>
<tr><td>OVLAY_MASK</td><td colspan="2">The display mode for the element overlay: 0= wireframe, 1 = hidden, 2 = shaded, 3 = current</td></tr>
<tr><td>OVL_R_MASK</td><td>Overlay red bits</td><td rowspan="3">It is assumed that a fully occupied bit field is 100% of the given component value for all these types</td></tr>
<tr><td>OVL_G_MASK</td><td>Overlay green bits</td></tr>
<tr><td>OVL_B_MASK</td><td>Overlay blue bits</td></tr>
<tr><td colspan="3">The following are D3PLOT-specific and reflect its internal storage of colour. External programmes would not normally use these, and can ignore them. They are included here for completeness.</td></tr>
<tr><td>OVL_CURRENT</td><td colspan="2">Whether element overlay uses "current" colour or some other</td></tr>
<tr><td>OVL_DEFAULT</td><td colspan="2">Whether element overlay uses the parent element colour</td></tr>
<tr><td>ENTITY_DEF</td><td colspan="2">Whether elements use their default parent colour</td></tr>
</table>

9. Using "Tools" Options

USING "TOOLS" OPTIONS

This section acts as a brief introduction to the commands in the top (**Main Menu**) box.

9.1. Introduction to Main Menu Commands

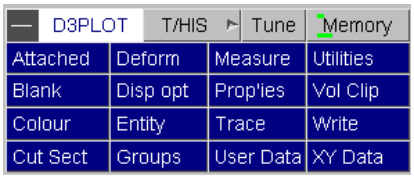
9.1.1. Commands Invoked from Here are Mutually Exclusive

Commands Invoked From Here are Mutually Exclusive

The commands in this box are mutually exclusive.

Panels for these commands are mapped in the fixed area below the main menu, stacked in order of invocation, and the button of the current command is highlighted.

Click on a command below to jump to its detailed description:



Attached	**Deform**	**Measure**	**Utilities**	**D3PLOT**	**T/HIS**	**Memory**
Blank	**Display Options**	**Properties**	**Volume Clipping**			
Colour	**Entity**	**Trace**	**Write**			
Cut Sections	**Groups**	**User Data**	**XY Data**			

9.1.2. Selecting Entities for Operations

Selecting Entities for Operations

Several of the functions in the main menu require you to input <lists> of entities to be processed. The standard procedure for this is:

(a) Select an entity type
(b) Define a <list> of these entities

In earlier versions of D3PLOT, all the menus used a display list for entity selection, see Using menus with "display list" selection .
From D3PLOT 14.0 onwards, some menus use object type for entity selection, similar to PRIMER, see Using menus with "object type" selection .

9.1.3. Using Menus with "Object Type" Selection

Using Menus with "Object Type" Selection

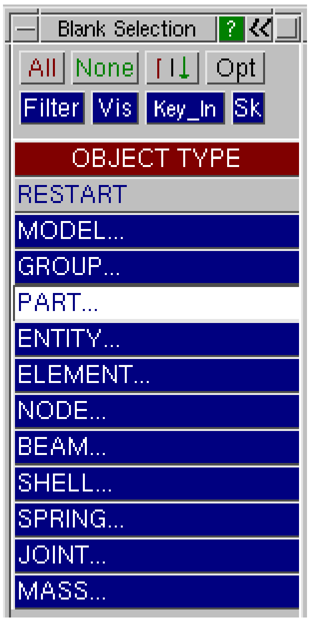
Selection takes place using a system of cascading "object menus", combined with screen-picking, area selection and keyed in data.

Primary selection of object type

In whatever context you are operating, here **BLANK** , you will be presented with the primary menu of object types to operate upon.

In this example the user has chosen **PARTS** from the range of possible categories. The list of categories available will depend on the operation being carried out, the model contents, and the window tabs.

To return to a different object type, click the **OBJECT TYPE** button. This will deselect all entities and display the primary menu of obejct types again.



Selection of objects from the menu list

Once an object category has been selected you are presented with a list of possible choices. In this case there are two models, each with several parts, and the user has selected one part from model 1 and two from model 2.

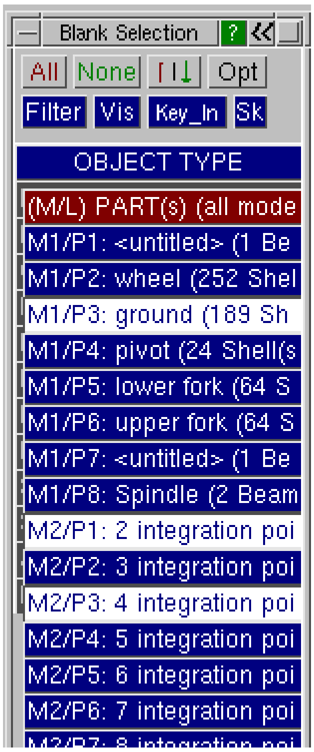
Objects can be selected or deselected (by clicking on them again) at will.

You can also use

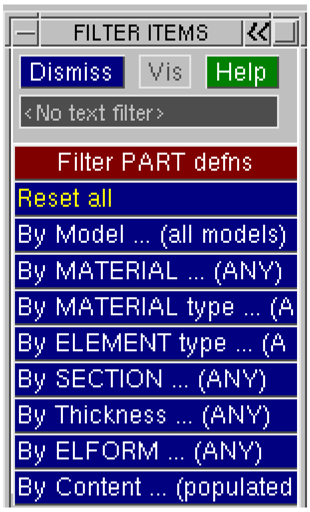
<table>
<tr><td>All</td><td>To select all eligible items</td><td rowspan="3">Note that All , None and I (nvert)
only operate upon what is shown in the menu
for reasons that will become apparent below.</td></tr>
<tr><td>None</td><td>To deselect all eligible items</td></tr>
<tr><td>I (invert)</td><td>To invert the current selection</td></tr>
</table>

Of the other buttons at the top of this panel:

Opt (ions)	Further options (refresh and blanking)
Filter	Applies "filtering" to what is shown
Vis	Maps a panel showing further "visible" picking options
Key_In	Maps a panel allowing you to key in label ranges directly
Sk (etch)	Sketches what is currently selected.



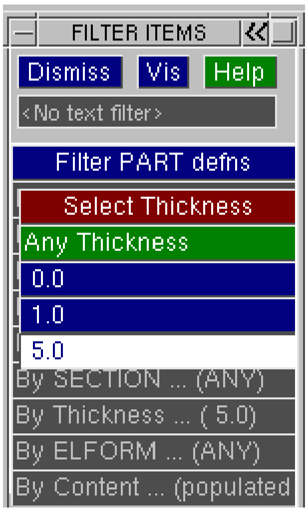
Using **Filter** to limit what appears in the menu



In some cases the list may be hundreds of items long, and a method of cutting down what is displayed is required. This is provided by the **FILTER** button at the top of this box.

This allows you to control what is displayed in the selection menu by providing a series of tests against objects are compared before they are included. The tests vary by object type, those for PARTS are shown here. By default all tests are unset (**ANY**), but you can set any combination: multiple ones combine in effect.

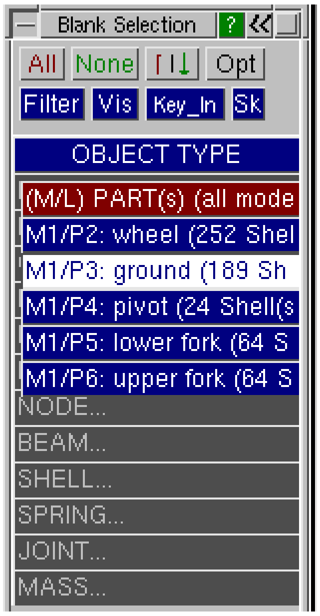
An example of setting a **Filter**



Here the **Thickness ...** option for PARTS has been selected.

You are presented with all thicknesses in the model(s), and can choose one. The **ANY** default may be chosen to revert to no filtering by this category.

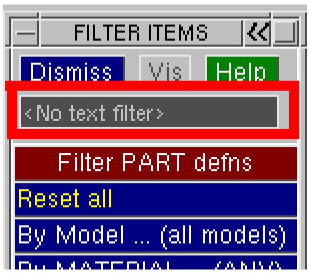
Some filter types are screen-pickable. If so, the **Vis** button in the filter menu is active. You can either select an explicit row, or use **Vis** and screen-pick from the current image.



In the example shown here the user has selected **5.0** as the thickness filter. This causes the selection menu to be updated immediately to show only those PARTs with thickness 5.0.

Note that the **ALL** and **NONE** options ***will now only operate on the five parts shown here.*** They will not affect the selection status of anything picked previously that does not now appear in the menu list.

Using Text Filter to search by text string



By typing something into the text filter box you will limit what is displayed in the main object menu to items that match that string.

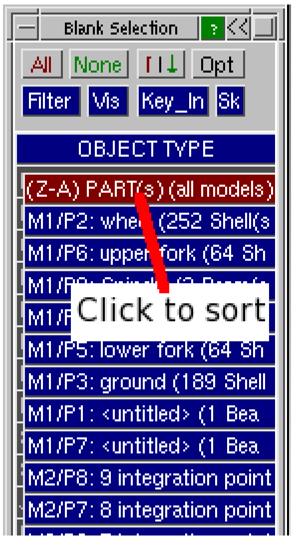
To cancel text filtering simply delete the contents of the box, and it will revert to showing <No text filter>.

Text matching is not case sensitive. White space at the beginning and endis ignored, however embedded white space in a string is considered when pattern matching. In addition filtering supports the following "wildcard" characters:

- **?** means match any single character
- ***** means match any number of characters.

These may be used any number of times in a string, for example " ***quick*fox*** " will match " **The quick brown fox jumped** ".
The string is considered to have ***** at its beginning and end, so typing in " **seat"** is equivalent to typing in " ***seat*** ".

Sorting the object menu contents



By default object menus are present in **M** odel/ **L** abel order, referred to as M/L, meaning

- All items in Model #1, sorted into ascending order by item label
- All items in Model #2, sorted by ditto
  etc

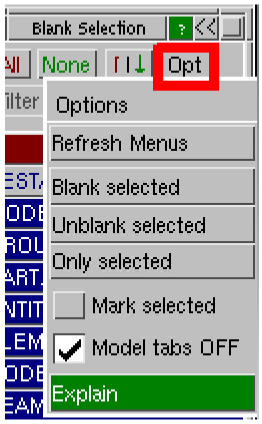
However it is possible to resort object menus dynamically by clicking on the menu title bar, which will cycle the sort process through:

- A-Z (alphabetic sorting by item title)
- Z-A (reverse alphabetic sorting)
- 0-9 (ascending numeric sorting by item label)
- 9-0 (reverse numeric sorting)
- M/L (default **M** odel/ **L** abel sorting)

Further clicks will repeat the cycle above.

The menu title row changes to show the current sorting method.

The Opt(ions) popup menu



Refresh Menus	Refreshes the current menu, updating it to reflect changes to things such as titles and set contents which may not have triggered an automatic menu refresh
Blank selected	Blanks the selected items
Unblank selected	Unblanks the selected items, turning on their entity visibility switches if necessary in order to make them visible
Only selected	Blanks everything except the selected items, again turning on their entity visibility switches if required.

When "Only" is used all other items in all models will be blanked, regardless of whether this object menu refers to a single model or multiple ones. This is necessary if "only" the selected items are to be visible.

9.1.4. Using Vis(ible) Screen Picking to Select Items

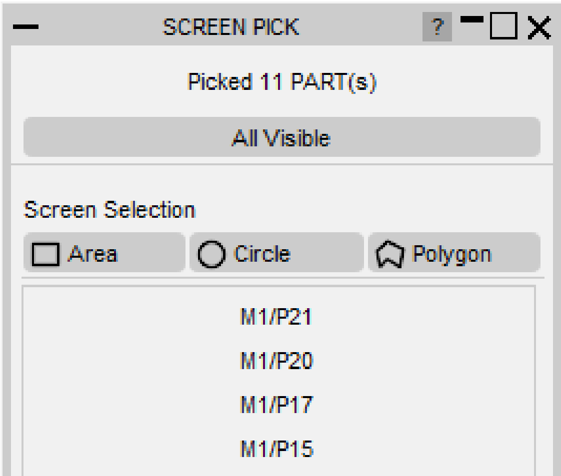
Using Vis(ible) Screen-Picking to Select Items

As well as selecting items from a menu you may pick them from the screen using the mouse. A combination of selection from the menu and screen-picking may be used, they are simply different ways of performing the same task.

Screen quick-pick is always "live" in the graphics window once you have selected an object category that is capable of being picked, it is not necessary to select **Vis** explicitly.

Through quick-pick there is:	
"Scalar" picking of single items	Just click on the approximate centre of the item to select it.
"Rectangular Area" picking of a range of items	Click and drag out a rectangular area. Everything within the area is selected.

Within the **Vis** panel there are the following further screen-picking options:	
All Visible	Will select automatically all "visible" items that are eligible. Note that "visible" in this context means what is displayed on the screen, but not necessarily what you can see. Items hidden behind other items are still "visible", as are items off the border of the current window. A more precise definition of "All Visible" would be "things which would be visible in a wireframe plot autoscaled to fit in the current window".
Screen Selection options:	
Area	Is an alternative way of defining a rectangular area by picking two points at opposite corners. Eligible items within the rectangle are selected.
Circle	Selects within a circular area. Click on the centre of the circle, drag out to define its radius and release to select eligible items within the circle.
Polygon	Selects within an arbitrarily shaped polygon. Select three or more points (up to a limit of 100) to define the polygon, and close it when complete. All eligible items within the polygon will be selected. While the polygon may be any shape, and include concave sections, it should not be excessively complex; and it is also recommended that it should not have crossed edges since while these will work the algorithm used to distinguish "inside" from "outside" may become confused by them.



9.1.5. Using Menus with "Display List" Selection

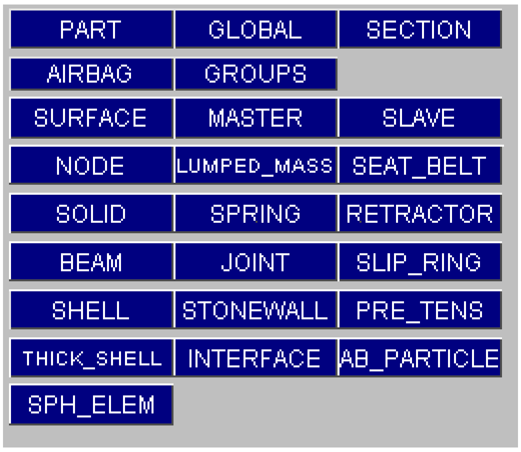
Using Menus with "Display List" Selection

Several of the functions in the main menu require you to input <lists> of entities to be processed using the display list menu style.

(a) Select an entity type

This figure shows the standard entity type entry panel. All possible entity types are always mapped, with those that are unavailable greyed out.

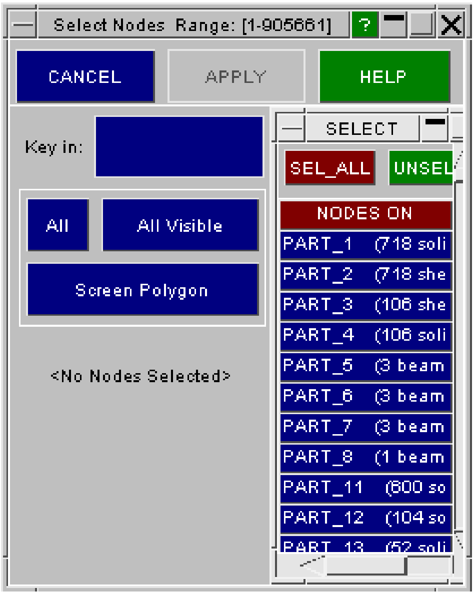
In some contexts some or all of the **MATERIAL** , **GLOBAL** , **SURFACE** , **SURFACE A & B** options shown here may not be present since they are not appropriate.



(b) Define a <list> of these entities

After you have chosen an entity type you must define a <list> of entities to be processed

This figure shows a typical panel displayed after a command. You can select a immediately by clicking on or dragging across visible entities. Other options are:



Key in	To type in a range;
ALL	To select all entities;
ALL_VISIBLE	All currently visible entities;
SCREEN POLYGON	Pick points defining a polygon within which entities will be defined

Or select entities on material(s) using the menu (here headed **NODES ON**).

9.1.6. Further Notes on Entity Selection

Further Notes on Entity Selection

"Hover over": showing what will be selected.

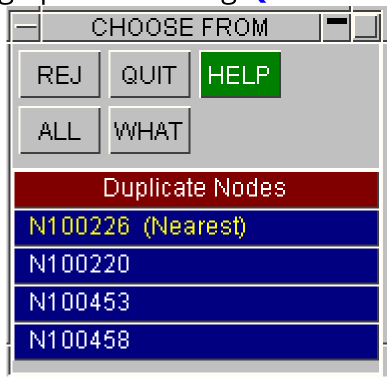
By default hovering the cursor over a row in an object menu will also highlight and label the item on the screen, helping to identify where in the model the item is.

Resolving ambiguous screen picks.

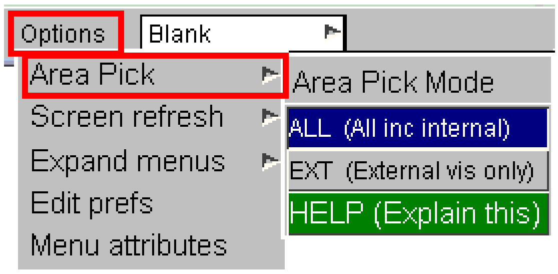
When screen-picking you may occasionally pick a point that does not lie unambiguously on an entity. In this case the "ambiguous pick" menu, see the figure (right), will be mapped, and you will be forced to choose an item. You can:

- Choose a menu item: the top one is always nearest to the point you picked, or:
- **REJ** ect this pick: the pick is ignored and you get another chance.
- Take **ALL** items from the menu list.

- Show **WHAT** these items are by labelling them.
- Abort the whole picking operation using **QUIT** .



Treatment of 3D elements during Area or Polygon picking

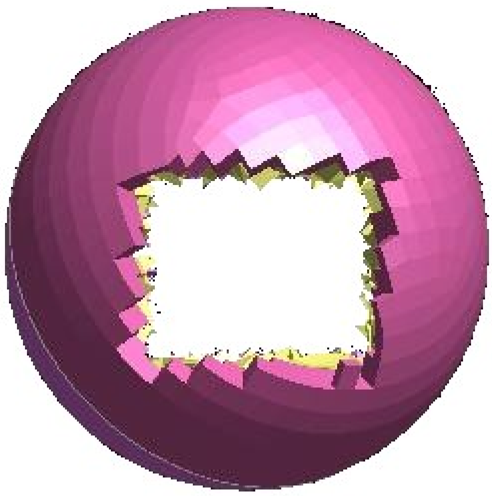


Options, Area Pick controls how 3D elements are treated during screen area type picking. (This does not affect single picks, which will always take the nearest element only.)

ALL All eligible elements in area

By default area picking of a mesh that contains solids or thick shells will include elements that are eligible for display, but which have not actually been drawn because they are interior to the mesh.

You can think of this as all elements in the "tunnel" behind the screen area: blanking in this way will punch a clear hole right through the mesh as shown in this example.



EXT Only external elements	Alternatively only those elements which have actually been drawn, for a 3D mesh the **EXT** ernal surface, will be selected: elements culled from the display because they are internal are not selected. The effect of a pick in this mode is like peeling an onion: only the outer layer is removed in each selection pass. Contrast this example with the image above: only the outer layer of solids has been removed.	

9.1.7. Using Command-Line Syntax for <Lists> of Entities

Using Command-Line Syntax for <lists> of Entities

In some circumstances it can be much quicker to use command-line input (in the dialogue box) to define <lists> of entities. A typical example might be when you want to process an explicit list of known elements, or a well defined range.

Typical command-line syntax to perform an operation is:

```
Command (command) (command) <list of entities>
```

for example:

```
/WRITE NODES 21 to 100
```

The valid syntax for a <list> of integers defining a range of entities is:

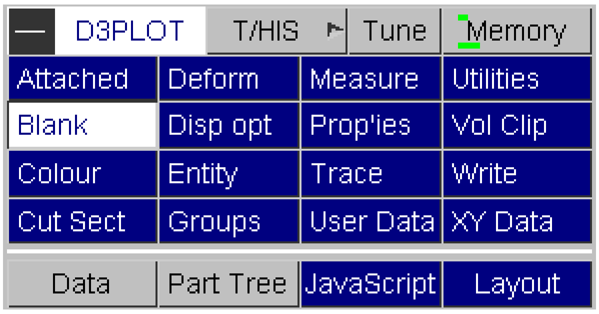
Single values	`12 20 200 -1`
a TO b (STEP c)	`1 TO 100,` `-20 TO 40 STEP 4,` `1000 TO 100 STEP -2`
All in context ALL or *	
Range limits	`FIRST` and `LAST`

All of these input types above may be mixed at will on a single line. Continuation lines, using `\` , may be used in the same way as for command words.

You can mix screen-menu and command-line input at will.

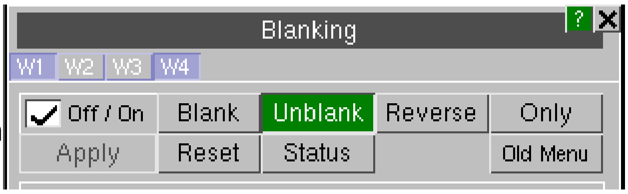
9.2. BLANK "Blanking" Controls the Visibility of Nodes and Elements

BLANK "Blanking" Controls the Visibility of Nodes and Elements



You can cut down what is displayed by "blanking" nodes and elements. (Unlike ENTITY display control blanking is selective: you can blank and unblank individual elements.)

Each node and element in your model has an internal blanking flag, which is initially set to **off** (i.e. the entity is visible). You can turn this flag **on** by blanking that entity, and in subsequent plots it will not be drawn until unblanked again.

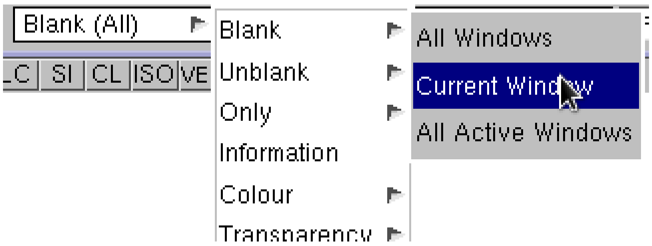


Blanking is "per window"

From D3PLOT 14.0 onwards, blanking can be done on a per window basis. If the same model is viewed in multiple windows, different parts of the model can be visible in different windows. This was not the case in earlier versions of D3PLOT.

In the blank menu, there are window tabs, allowing users to set which windows the blank menu acts on.

DYNAMIC ("Quick Pick") BLANKING



The **BLANK** panel is used when you want to exercise detailed control over what is blanked. The alternative method of blanking, referred to as "quick pick" mode, which is more suitable for simple blanking is described in "QUICK PICK" Options.

In the quick pick menu, under **BLANK** there is a popup with the option to blank in **All** Windows, the **Current** Window, or All **Active** Windows. The default is **All** Windows. The image deomonstrates changing to the **Current** Window.

Likewise, there are similar popups for **UNBLANK** and **ONLY** . For **UNBLANK** rather than **Current** Window the option is Only Window [ ] with a text box to enter the desired window number. This allows the users to select items to unblank which are visible in another window.

9.2.1. The BLANK Menu

The BLANK Menu

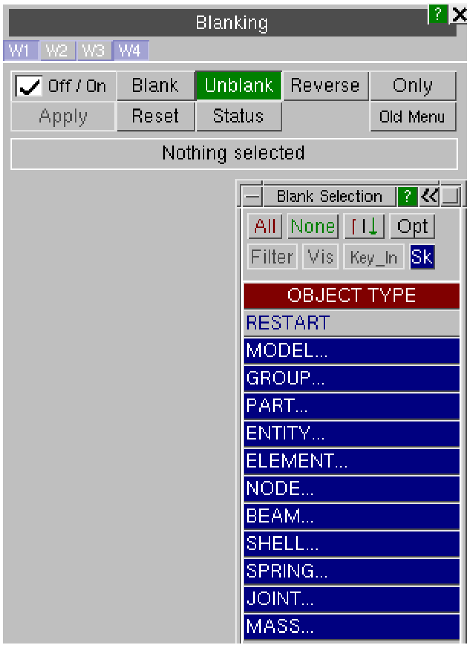
Use **OFF/ON** to toggle whether blanking is applied in the active windows.

The blank menu can be used in four different modes:

BLANK	To blank the selected items
UNBLANK	To unblank the selected items
REVERSE	To reverse the blanking status of the selected items
ONLY	Blank all items in the active windows, leaving only the selected items

Further buttons on the blank menu:

RESET	Deselects everything in the Blank Selection menu and clears all filtering.
STATUS	Lists how many items are currently selected in the menu.



Items are selected using an "object type" selection menu (see Using menus with "object type" selection for details).

For example, click on "Part" to get a list of all the Parts in the model. Then select some parts to apply the blanking operation to.

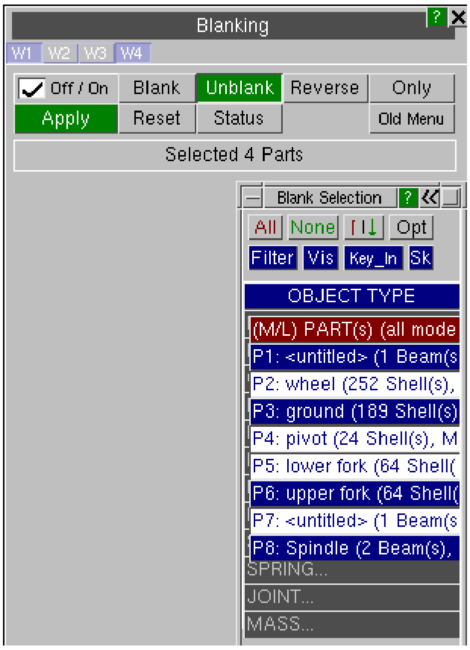
It is possible to select these parts from the list, **Filter** the list, use **Vis** ible screen picking, or **Key_In** an ID (see Using menus with "object type" selection).

Once some items are selected, the **APPLY** button is active.

Use **APPLY** to apply the operation. By default, the image is updated after **APPLY** has been clicked.

Note, the user may change the blank menu mode once the items are selected. For example, having applied a blank to some parts, one might change to unblank mode and apply an unblank to the same selection of parts.

Note, changing to a different object type will reset the current selection. You should **APPLY** your blank operation for each object type.



9.2.2. The Old BLANK Menu (D3PLOT 13.0 and Earlier)

The Old BLANK Menu (D3PLOT 13.0 and Earlier)

The BLANK panel operates on one model at a time.

If you have more than one model then you will be forced to choose which one the blanking panel operates on. You can subsequently change this by using the `M1 .. Mn` tabs.

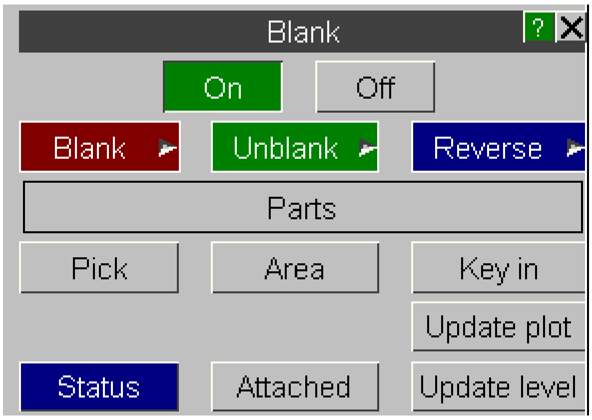
Blanking is "per model", not "per window"

The blanking status of items is stored at the model level, not at the window level. Therefore if something is blanked in one window it will also disappear from any other windows in which it occurs once they are redrawn.

This is done for simplicity and to save memory. If you need to have two concurrent images of a model with different blanking attributes then you will need to read that model in twice, treating it as two separate, albeit identical, models.

The **BLANK** menu is split into 2 sections.

Selecting Items



The top half of the menu provides ways to select the items to be blanked and unblanked while the bottom half is used to choose the type of item that is going to be blanked / unblanked.

ON	Turns blanking ON so that blanked items are not drawn.
OFF	Turns blanking **OFF** so that all items are drawn even if they have been blanked. This option does not reset the blanking status of items so that when blanking is turned **ON** again items that were previously blanked are still blanked.

These options can be used to modify the blanking status of a complete category of elements. In addition these option can also be used to **BLANK** / **REVERSE** / **UNBLANK** the whole model.

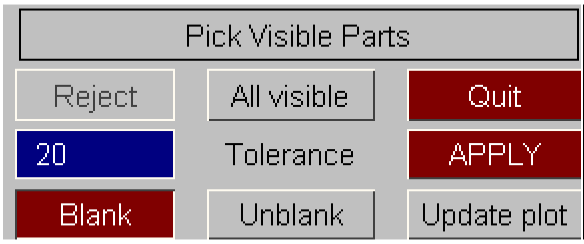
PICK	Pick items individually to be blanked / unblanked, (see below for more details)
AREA	Pick items by area to be blanked / unblanked (see below for more details)
KEY IN	Type in the ID of items to be blanked/unblanked
UPDATE PLOT	Redraw the image
STATUS	Lists the blanking status of all items

ATTACHED With this option selected any item that shares a node with an item that is picked is also blanked or unblanked along with that item

UPDATE LEVEL The UPDATE LEVEL controls whether items are blanked dynamically. If the **UPDATE LEVEL** is set to 3 then dynamic blanking is turned on.

PICK Pick items to be blanked / unblanked

If DYNAMIC blanking is turned on items are blanked as they are picked. When picking items the mouse buttons have the following function:



Left Mouse Pick an item

Middle Mouse Reject the last item selected (Update Level 1 & 2)

Right Mouse Deselect an item (Update Level 1 & 2)

Other Options :

REJECT Reject the last item selected (Update Level 1 & 2)

ALL VISIBLE Select all items currently visible on the screen. Items outside the curtrent screen area are not selected.

QUIT Quit without blanking/unblanking selected items (Update Level 1 & 2)

TOLERANCE Define a screen tolerance for picking items.

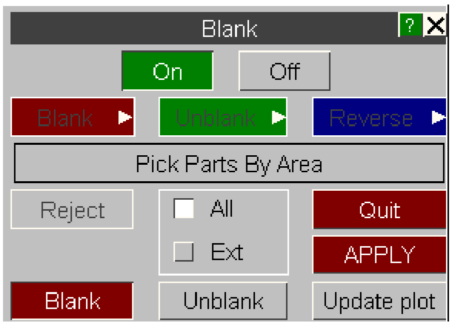
BLANK Items that are selected are blanked

UNBLANK Items that are selected are unblanked

APPLY Blank / Unblank selected items (Update Level 1 & 2) and then return to the main BLANK menu (Update Level 1,2 & 3)

UPDATE PLOT Redraw the image with the currently selected items blanked / unblanked.

AREA Pick items by area to be blanked / unblanked



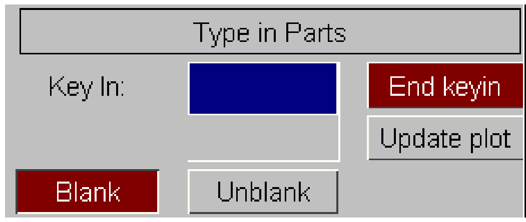
If DYNAMIC blanking is turned on items are blanked as they are picked. When picking items the mouse buttons have the following function.

Left Mouse	Define a rectangle and select items within it.
Middle Mouse	Same as **Left Mouse**
Right Mouse	Define a rectangle and deselect items within it. (Update Level 1 & 2)

Other Options :

REJECT	Reject the first point selected
ALL or EXT	For 3D elements (e.g. Solids) selection can be applied to ALL elements or only EXTernal elements in the area.
QUIT	Quit without blanking/unblanking selected items (Update Level 1 & 2)
BLANK	Items that are selected are blanked
UNBLANK	Items that are selected are unblanked
APPLY	Blank / Unblank selected items (Update Level 1 & 2) and then return to the main BLANK menu (Update Level 1,2 & 3)
UPDATE PLOT	Redraw the image with the currently selected items blanked / unblanked.

KEY IN Enter the ID of items to be blanked / unblanked



END SELECTION Return to the main **BLANK** menu.

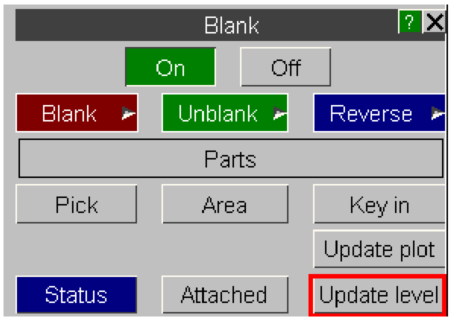
BLANK Items that are selected are blanked

UNBLANK Items that are selected are unblanked

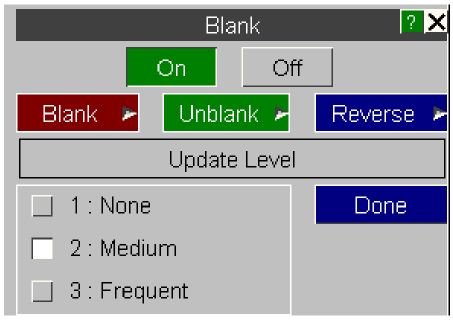
APPLY **Blank** / **Unblank** selected items (Update Level 1 & 2) and then return to the main **BLANK** menu (Update Level 1,2 & 3)

UPDATE PLOT Redraw the image with the currently selected items blanked / unblanked

UPDATE LEVEL Image update frequency



The top half of the menu provides ways to select the items to be blanked and unblanked while the bottom half is used to choose the type of item that is going to be blanked / unblanked.



1 (NONE) The image is never updated automatically to show the effect of blanking. You must redraw the image (eg DR, SH, etc) to see the effect of changes. This is not recommended unless you have a very slow connection to your display.

2 (MEDIUM) Default behaviour. (Un-)Blanking something does not cause the display to update, but any subsequent viewing operation (eg zoom, dynamic view, etc) will result in the image being updated to show the effect of changes.

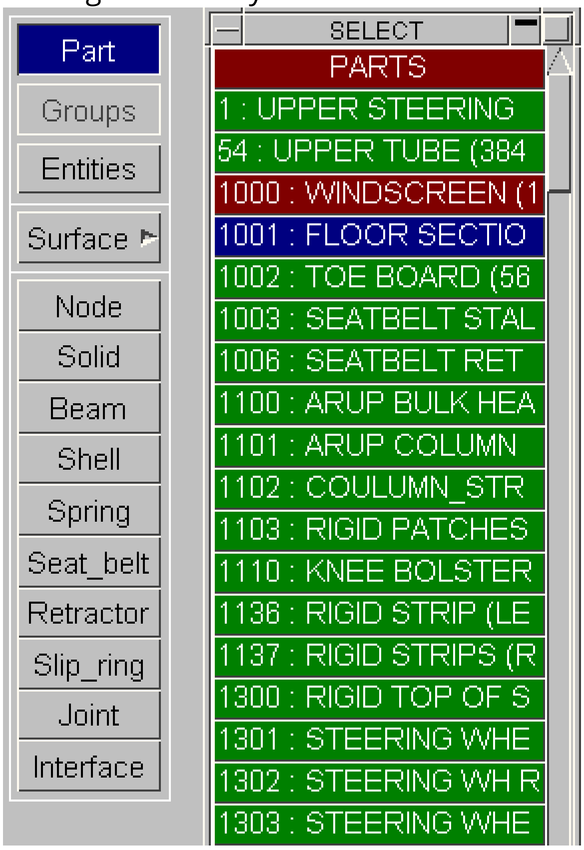
3 (FREQUENT) Immediate update. Every time you (un-)blank something the display will be redrawn immediately to show the effect of the change. Very large models on slow displays may become cumbersome if this is used.

It is recommended that you keep the default **UPDATE LEVEL** of **2** , and use the **UPDATE PLOT** button explicitly to see the effect of changes. (The " Quick Pick " blanking option provides "instant" blanking, and is a better way of achieving this behaviour.)

Entity Types

The bottom section lists the generic entity types that the model contains and this section controls the type of items that are selected by the **PICK** and **AREA** and **KEY IN** options.

By default this section shows a list of all the PARTs that the model contains. The blanking status of an item can be changed by clicking on its entry in the list. The list of items is colour coded as follows :



RED	The whole of the item is blanked.
GREEN	The whole of the item is unblanked.
BLUE	Some of the item is blanked.

If a BLUE menu entry is clicked on the item will be completely blanked. Clicking on the menu entry a second time will then completely unblank the item - **it is not possible to return to the partially blanked state.**

If an option other than PARTS is selected then the list is automatically updated to list the appropriate items. To reduce the number of generic entity types the **SEATBELT** and **SURFACE** buttons are linked to popups containing related entity types.

If DYNAMIC blanking is active then items are blanked / unblanked as they are selected in the list. If DYNAMIC blanking is not active the menu will be updated as items are selected but the image will not be updated until either the view is changed (rotated, zoomed etc) or the image is explicitly redrawn (**HI** , **CT** , **SH** etc).

9.3. VOLUME_CLIPPING

VOLUME_CLIPPING

By default no volume clipping is in effect, and pressing the **VOLUME CLIPPING** button will give the main panel in its basic state, as shown in the figure (right).

9.3.1. Volume Clipping is a "Per Window" Attribute

Volume Clipping is a "Per window" Attribute

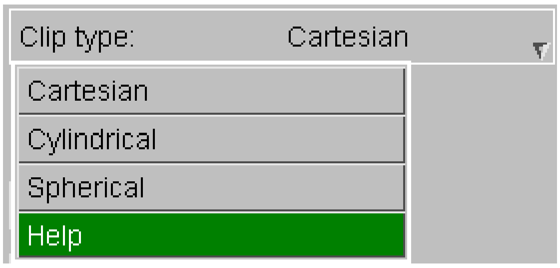
Volume clipping definitions apply to all those windows which have their `W1 .. Wn` tabs set. Clipping takes place in the specified space system in each window, and will apply to all models in that window.

9.3.2. CREATE Creating a New Clipping Volume

CREATE Creating a New Clipping Volume

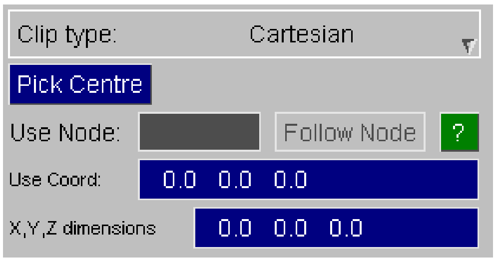
When creating a volume you need to define its type.

The options are **Cartesian** , **Cylindrical** and **Spherical** which can be picked from the popup menu as shown on the right.



Defining a **Cartesian** volume

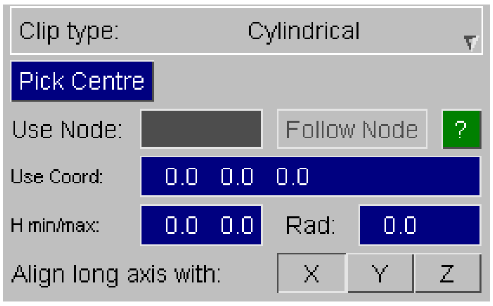
A cartesian volume is defined by:



Centre point:	Use **Pick Centre** to select a nodal coord or type in at **Use Coord**.
The X, Y, Z dimensions	Type in X, Y, Z dimensions

Defining a **Cylindrical** volume

A cylindrical volume is defined by:



Centre point:	Use **Pick Centre** to select a nodal coord or type in at **Use Coord**.
Height:	Type in the maximum and minimum height at **Hmin/max**:

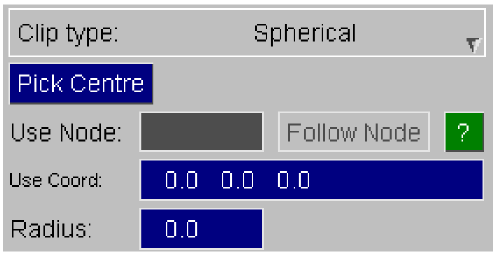
Radius:

Type in at **Rad:**

Align the long (height) axis on one of the global X, Y or Z axes with the relevant button.

Defining a **Spherical** volume

A spherical volume is defined by:



Centre point:

Use **Pick Centre** to select a nodal coord or type in at **Use Coord**.

Radius:

Type in at **Radius:**

Follow Node

Normally the volume will stay in the same postion through the animation. If the volume centre has been selected by specifying a Node you can set the volume to move with it through the animation.

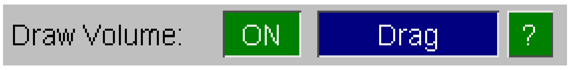
NOTE: This option is not available if the space system selected is BASIC, since this always uses the undeformed position.

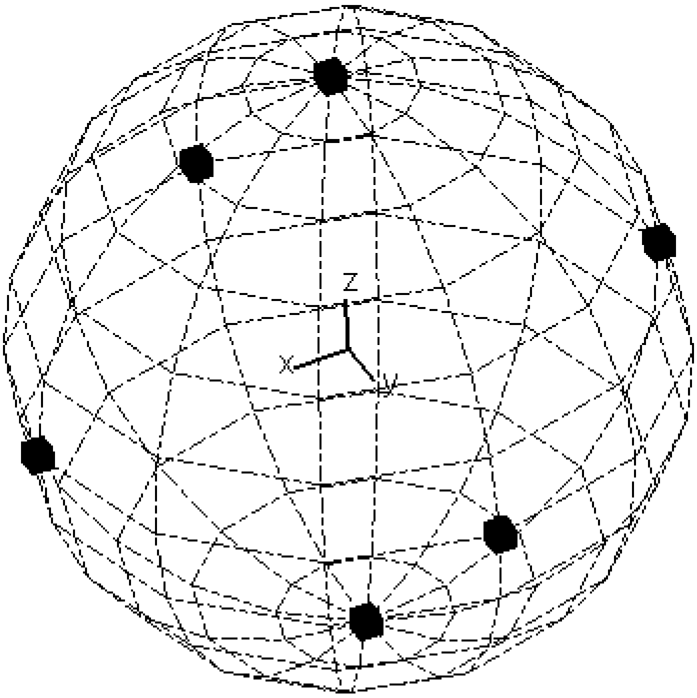
9.3.3. DRAG Resize and Reposition the Volume

DRAG Resize and Reposition the Volume

Once a volume has been created you can resize and repostion it by dragging it on the screen. To do this turn on the **Drag** button.

The volume will be drawn on screen with 'handles' that can then be dragged to resize it:





To reposition it press and hold:

LEFT mouse button:	Translate in global X direction
MIDDLE mouse button	Translate in global Y direction
RIGHT mouse button	Translate in global Z direction

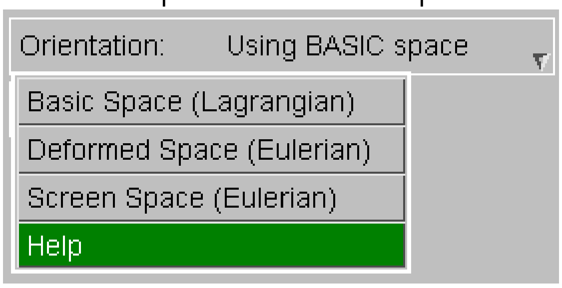
9.3.4. ORIENT Defining a Space System for Volume Clipping

ORIENT Defining a Space System for Volume Clipping

When you create a volume, you must define a space system for it. This figure shows the space system definition panel.

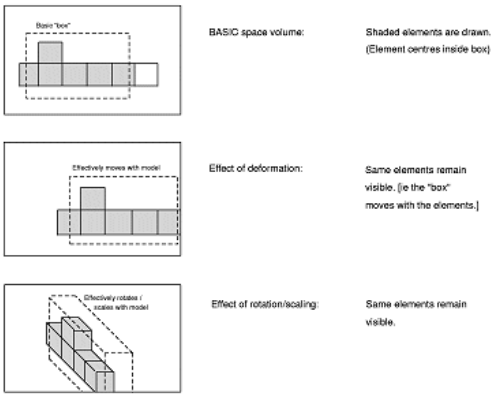
There are three options: **BASIC** , **DEFORMED** and SCREEN space which can be picked from the popup menu as shown on the right.

These have the following meanings:



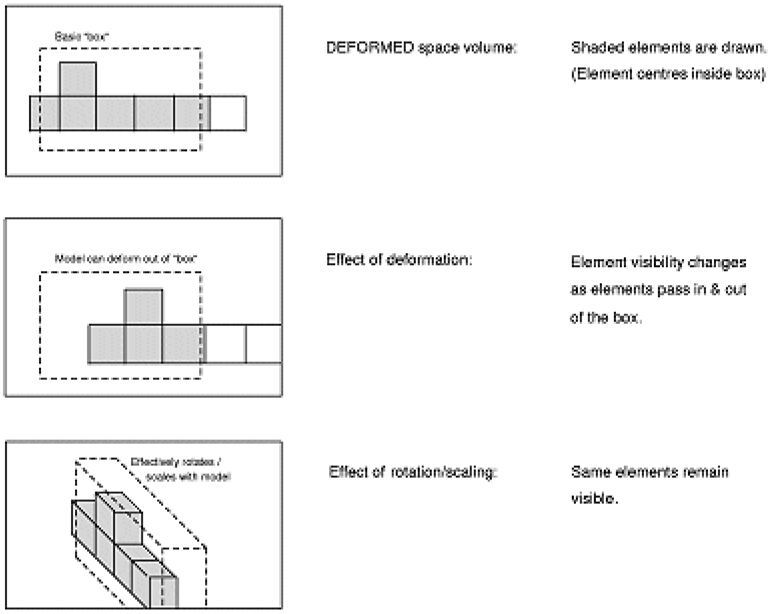
BASIC

Clipping is based on undeformed nodal geometry. So the same elements are always visible regardless of their deformations or any changes of view.



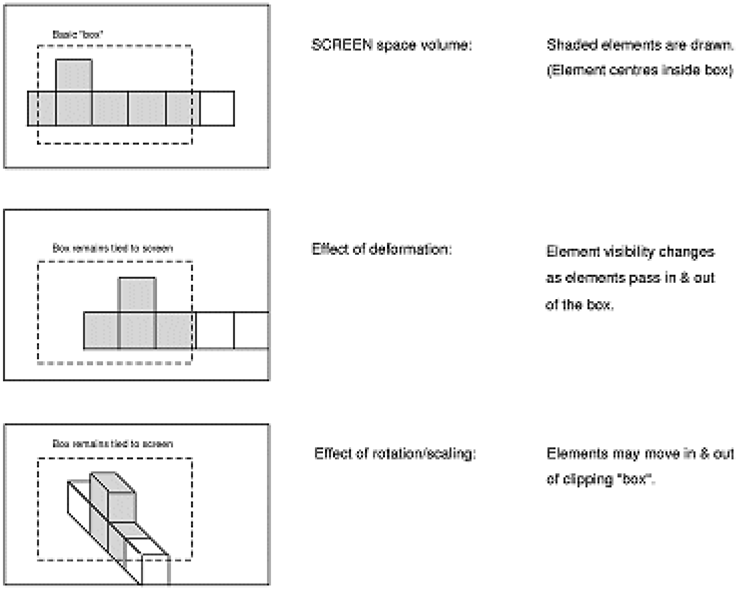
DEFORMED

Clipping is based on the deformed nodal geometry at each state. So elements may pass in and out of the clipping volume as they move and deform.



SCREEN

lipping is tied to screen coordinate space. Thus rotation and scaling operations, as well as deformations, may move elements in and out of the volume



9.3.5. Other Actions

Other Actions

There are some other actions which can be applied:

Draw Volume Sketching the clipping volume

This button will draw the volume on the screen.

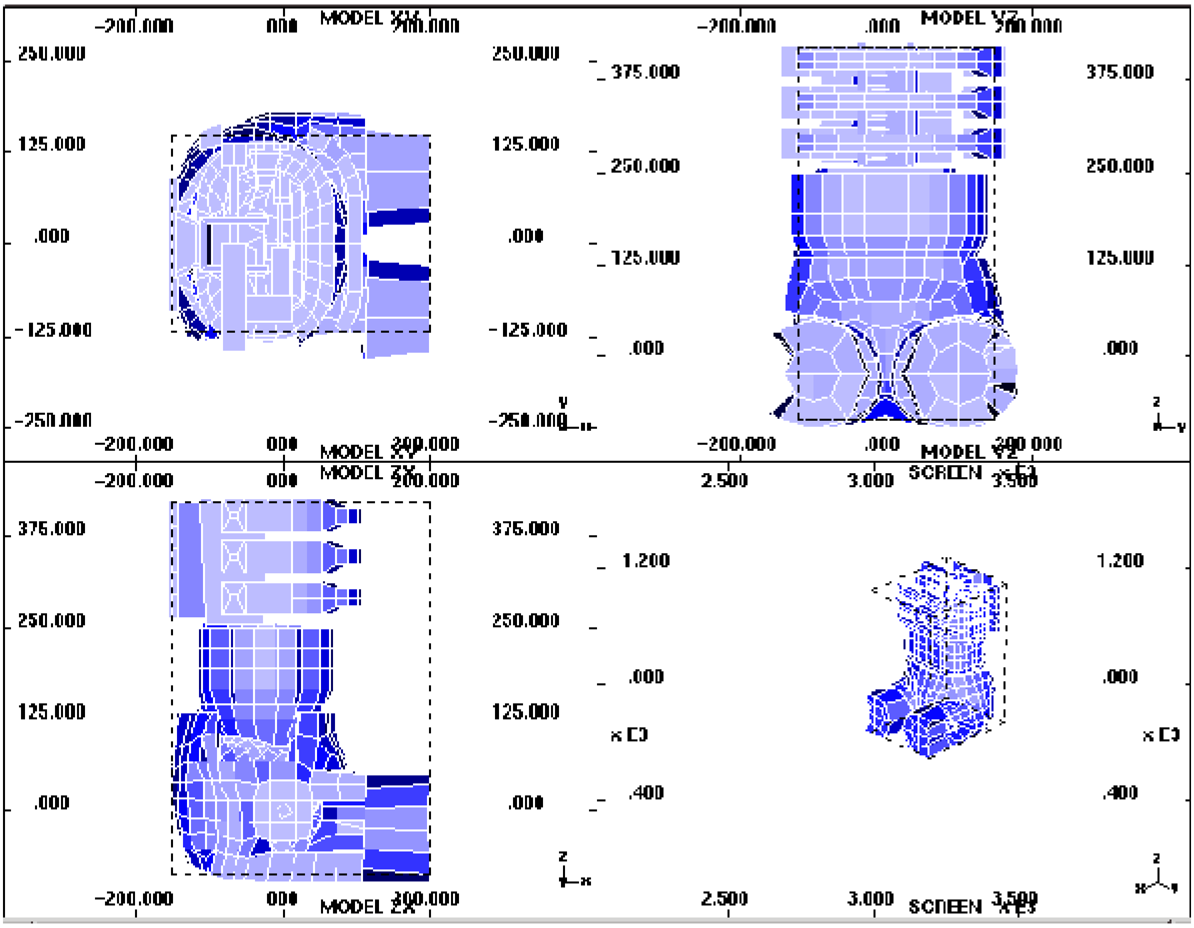
Discard what Discarding entities inside or outside the volume

By default entities OUTSIDE the volume are discarded, but you can invert the effect so that everything **INSIDE** the volume is discarded instead.

Location Plot Location plots showing 4 views of the current volume

This figure shows the image above, with the volume-clipping switch turned on, drawn as a "location" plot. The display mode used is that of the most recently issued drawing command.

This draws 3 standard views (on **XY** , **YZ** and **XZ**), and also the current view in the bottom right quadrant. The **GRATICULE** (see DISPLAY_OPTIONS) is also turned on to give you numeric feedback.



9.3.6. SAVE/RETRIEVE Managing the Storage and Retrieval of Clipping Volumes on Disk

SAVE/RETRIEVE Managing the Storage and Retrieval of Clipping Volumes on Disk

There is only ever one "current" clipping volume definition, but up to 100 such definitions can be stored in an external " `volume.clip` " file, and any number of such files may exist.

Volumes are model-independent and may be shared between dissimilar analyses

Storing and retrieving clipping volumes :

This figure shows the storage and retrieval sub-menu. The four commands in the left hand column manipulate volumes as follows:

STORE	Stores the current volume definition in the file.
GET	Reads a stored definition which overwrites the current one.
RENAME	Renames a stored definition.
DELETE	Deletes a stored definition.
FILE...	Lets you enter a new " `volume.clip` " filename: Any filename is permissable, but `volume.clip` is assumed, and the extension " `.clip` " is recommended (but not mandatory). Note that `volume.clip` files are binary, and are not normally transferrable between different machine types. Nor will you be able to read or edit them. However transfers between typical workstations (using IEEE format) will usually work OK.

Only **GET** affects the current definition, the other commands leave it unchanged.

All storage and retrieval operations take place using the current " `volume.clip` " file. If such a file has not been opened explicitly a file called volume.clip is opened automatically (and an empty file of this name is created if it doesn't already exist.)

You will note that volumes are stored with names as well as numbers. These are optional, but help when identifying which volumes does what. A maximum of 40 characters is permitted for each volume name.

9.3.7. Further Notes on Volume Clipping

Further Notes on Volume Clipping

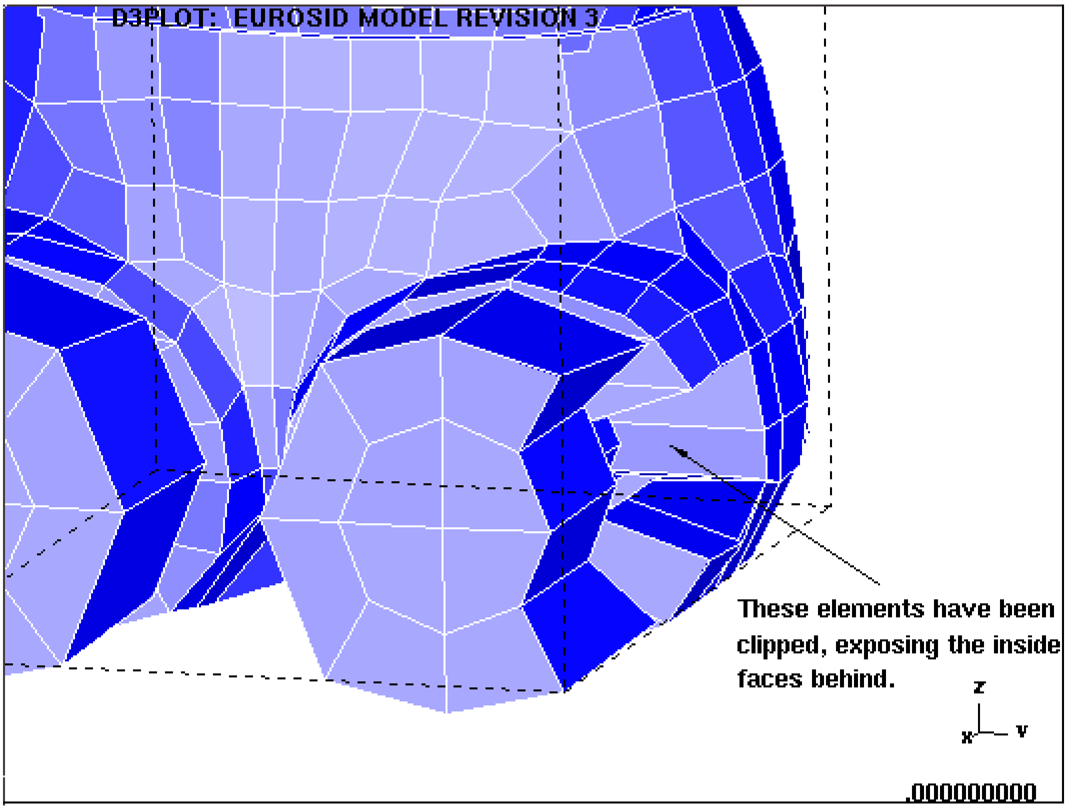
Note 1:	Clipping is calculated using the simple test: "is the element centre within the current volume?". Then either the whole element is displayed, or it is not drawn at all. No interpolation across elements takes place. This can give an effect rather like taking bricks out of a wall as shown in this figure.
Note 2:	Volume clipping does not work with stonewalls. This is because their geometry is at best strange, and often has infinite dimensions, making it too hard to implement.

Note 3: On 3D devices the graphics mode will be switched temporarily back to 2D when creating "location" plots.

Note 4: You cannot screen-pick entities from "location" plots.

Note 5: When a clipping volume exists you will find that the CV (Current Volume) button is live in entity <list> entry panels. (See Selecting Entities for Operations , and its accompanying figure .) This provides the option of selecting entities within the current volume.

Note 6: Volume clipping can affect other parts of D3PLOT, as shown in the following table:



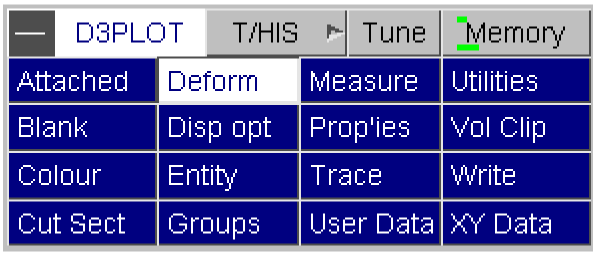
Function affected by clipping	Relevant D3PLOT command	Section
Averaging of element data at nodes	**AVERAGING... Clipping ignored switch**	AVERAGING... Controlling data averaging across adjacent elements
Calculation of free edges	**DISPLAY_OPTIONS FREE_EDGES... CLIPPING -> EDGES**	FREE_EDGES... Menu: Controlling Free Edge Display of Element Borders

Exclusion of elements from **WRITE SCAN** [WRITE] SCAN

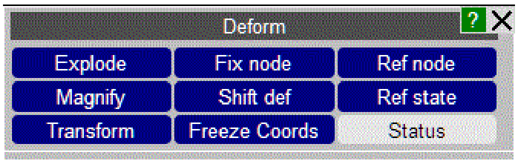
9.4. DEFORM Deforming Geometry

DEFORM Deforming Geometry

The **DEFORM** command contains four functions which "deform" geometry in various ways: exploding parts, magnifying displacements, fixing a node in space and fixing the model in space.



This figure shows the generic **DEFORM** panel, which gives access to its functions.



EXPLODE_PARTS	Artificially separates parts by applying "explosion" vectors to them
MAGNIFY_DISPLACEMENTS	Allows scales other than 1.0 to be applied to graphical displacements.
FIX_NODE	Subtracts the displacement at a node from that at all others, effectively "fixing" it in model space.
SHIFT_DEFORMED	Fixes three nodes, forming a local coordinate system, against which all displacements are drawn.
REFERENCE_NODE(s)	Makes results relative to those at one or three nodes
REFERENCE STATE/MODEL	Makes results relative to a "reference" state in this or another model
TRANSFORM	Apply translation, reflection, rotation and scale to a model as it is read in
FREEZE COORDS	Define a state from which the coordinates are used to display the results from all other states on.

9.4.1. DEFORM Options Apply at a Mixture of "Per Window" and "Per Model" Levels

DEFORM Options Apply at a Mixture of "Per Window" and "Per Model" Levels

All the options will apply to the windows selected by the `W1 .. Wn` tabs, but wherever node labels are used the following rules apply:

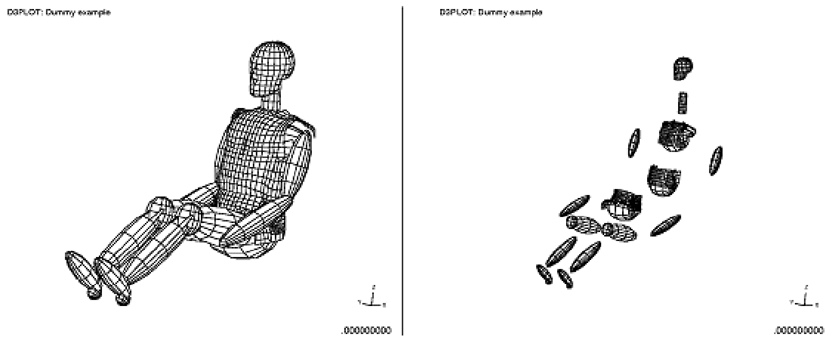
- Node labels will be mapped onto all relevant models.
- If a node does not exist in a particular model then that feature will be disabled in that model.
- If you screen-pick nodes you have to say which model they are to be picked from, but once picked the "label" rules above apply.

An exception is that the **TRANSFORM** option always works on a per-model basis.

9.4.2. EXPLODE_PARTS Separating ("Exploding") Parts

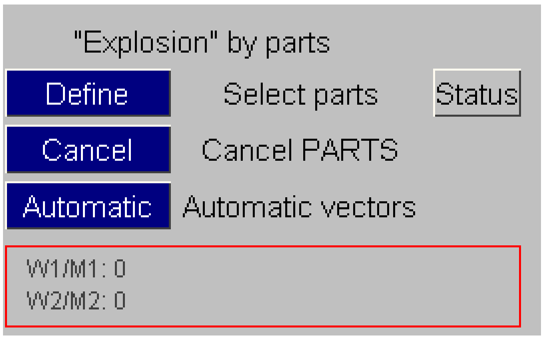
EXPLODE_PARTS Separating ("Exploding") Parts

"Exploding" a part is done by applying a [dx,dy,dz] vector to all nodes of that part, which has the effect of moving it bodily to a new location. By default no explosion vectors are set, but you can define, modify and cancel vectors for any part(s) at will. The figure below left shows an unexploded, and below right, an exploded dummy model:



This figure shows the basic "Explosion" control panel in its initial state.

In this example no explosion vectors have been defined yet, as can be seen in the status feedback area.

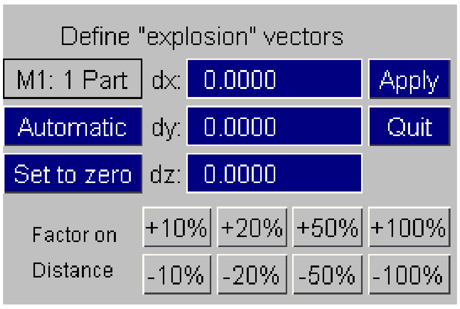


DEFINE Defining explicit explosion vectors for a <list> of parts

If you know exactly which parts you want to "explode", and by how much, use **DEFINE**. This requires you to define a <list> of parts, using the standard selection panel, then for each part it gives you the explosion vector definition panel shown in the following figure:

For each part you can define:

dx , **dy** , **dz**	Explicit vectors;
AUTOMATIC	Let D3PLOT calculate vectors (based on vector from model C.of.G to part C.of.G).
SET TO ZERO	Reset all 3 vectors to zero.
Factors...	Provides a simple way to factor the current vectors by known %age amounts.



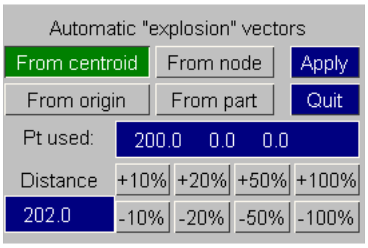
You can use the definition methods above in any order: for example use **AUTOMATIC** to get an initial estimate, then **Factor** them, or modify them by hand. The vectors are only stored when you give the **DONE** command. The vectors you define will take effect the next time you issue a plotting command.

CANCEL Cancelling (resetting to zero) explosion vectors

Explosion vectors remain in force until you change or **CANCEL** them explicitly. Use **CANCEL**, then select a <list> of parts to have their vectors zeroed. The effect will be seen the next time you issue a plotting command.

AUTOMATIC Automatically generated vectors for a <list> of parts

Typing in vectors for a long list of parts can get tedious, so it is possible to get D3PLOT to generate vectors for you automatically. These are based on the vector from a defined position (by default the centre of the model) to the centre of gravity (C.of.G) of each part, multiplied by a known factor.



This figure shows the **AUTOMATIC** vector definition panel. You define the centre from which vectors are calculated from one of:

CENTROID	Model centre of gravity
NODE	Nodal coordinate
ORIGIN	Coordinate [0,0,0]
MATL	C.of.G of a part
Pt used	Type in a coordinate

Then define a (non-zero!) **Distance** , or apply a %age factor to the existing value.

When you have defined the centre of explosion correctly press **APPLY** , and you will be asked to define a <list> of parts to which to apply vectors. Vectors for each part will be calculated based on the distance from the part C.of.G to the centre defined here, factored in proportion to the **Distance** value. The vectors generated are not usually ideal, but they provide a good starting point from which they can be "tweaked" to give the required image.

Notes on explosion vectors

Note 1: Vectors only affect plots, they have no influence on nodal coordinates used for X-Y plotting, or in **WRITE** , or upon the calculation of element volume etc.

Note 2: Explosions are applied as vectors added to nodal coordinates. Where two parts share a common node there is an ambiguity: should vectors be cumulative or, if not, which value whould be used? In this case D3PLOT uses the vector of the lowest numbered part, and ignores the remainder. So try to avoid this situation or, if you cannot, be prepared for elements on the border of two parts with dissimilar vectors to appear to be stretched.

Note 3: Explosion can be used in conjunction with the other options in the **DEFORM** menu. Transformations to nodal coordinates are applied in the order:

FIX_NODE or **SHIFT_DEFORMED** (Mutually exclusive)

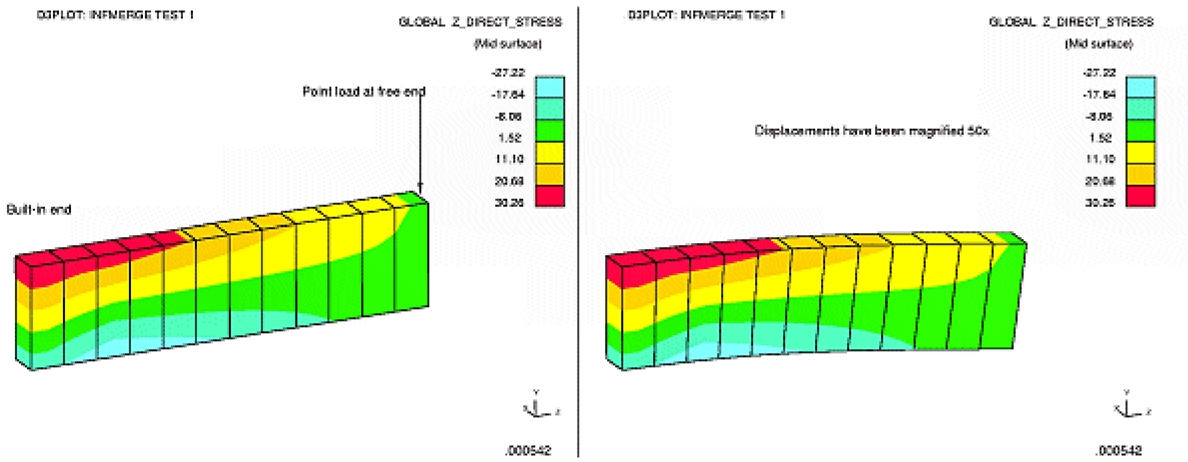
MAGNIFY_DISPLACEMENTS

EXPLODE

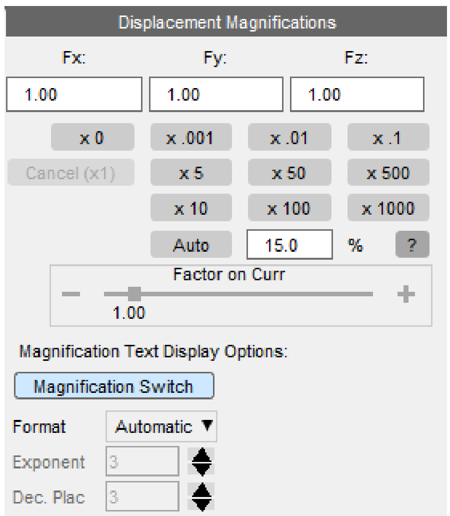
9.4.3. MAGNIFY_DISPLACEMENTS Factoring Nodal Displacements

MAGNIFY_DISPLACEMENTS Factoring Nodal Displacements

By default D3PLOT plots nodal coordinates at their true positions: a displacement factor of 1.0. However there are times when you may wish to factor displacements, for example when stresses are in the linear elastic range and displacements hardly visible. The figure below left shows an example of unfactored, and below right, factored (magnified by 50), displacements for a cantilever subject to a point load on its end.



General Displacement Magnification Factors



The **MAGNIFY_DISPLACEMENTS** value control panel is shown in the figure (right).

To enter factors type in [**Fx,Fy,Fz**], or use one of the pre-programmed factors (**x5** , **x10** , etc). The **CANCEL** button sets all factors back to the default of 1.0. The **Factor on Curr** slider applies the given factor to the current values - an easy way of setting any value.

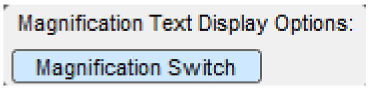
When you have defined the factors use **DONE** to return and apply them.

Factors take effect the next time you update the plot, and stay in effect until changed again.

Note 1: Factors only apply to plots. They have no effect on written of X-Y data output, calculation of volumes etc, or contoured values of displacement.

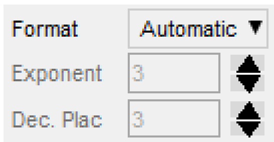
Note 2: Magnified displacements may be used in conjunction with the other **DEFORM** options: see Note 3 in Notes on explosion vectors for the order of operations.

Magnification Text Display Switch



The **Display Magnification Switch** button allows users to toggle the visibility of the magnification text in the graphics window. It should be noted that even if the switch is on, if a magnification is not set (i.e. the values are all 1 in all directions), the value will not be shown for most models. The exception includes cases such as when a Nastran OP2 model is loaded in the same Window at the same time. An alternative switch to toggle the visibility of these values is under **Display Options → Window Dressing**

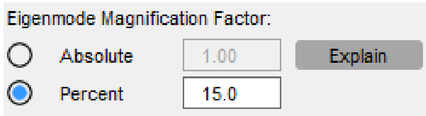
Magnification Formatting Controls



Additionally there is a dropdown box with formatting options **Automatic**, **Scientific**, **General** and **Manual**. **Automatic** changes the formatting of the numbers based on predetermined rules for each set of values. **Scientific** and **General** allow the user to customise the number of exponent, and **Manual** allows full control over the number of decimal places and the exponent value.

Eigenmode Magnification Factors

Magnifications set when reading in eigenmode models (d3eigv, Nastran OP2 and OptiStruct) in the OPEN PLOT FILE panel can be modified at a later stage in this panel. The main difference is that here, the settings are a per-window attribute (changes here affect each model in the window) compared to being per model at the file opening stage.



The magnification value specified in this panel overrides the value set in the OPEN PLOT FILE panel. The two values are not both applied to the model in a compound way.

9.4.4. FIX_NODE Fixing a Node Position Despite Displacements

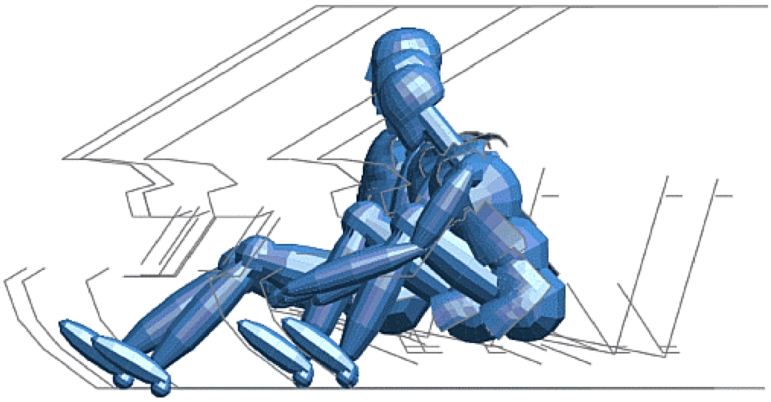
FIX_NODE Fixing a Node Position Despite Displacements

In some circumstances a model may move a long way between successive states, and it can be inconvenient to have it progressively disappearing off the screen.

This figure shows a few frames during the assembly of an animation of a dummy sled test analysis.

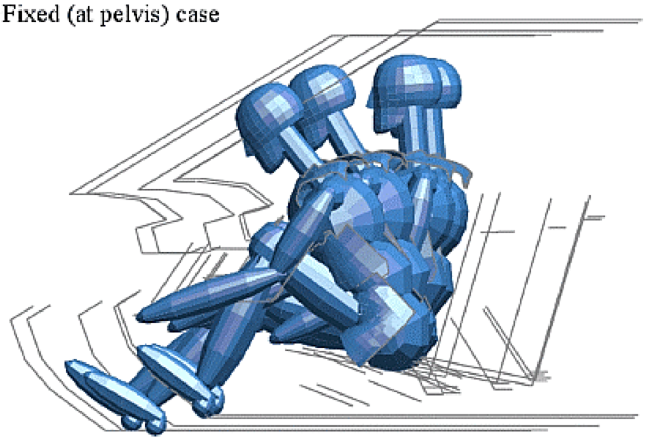
In these tests the sled is pulled backwards to mimic the deceleration during a crash, and it moves off the screen as a consequence. So a simple translation to bring it back to the undeformed position will suffice.

Normal (unfixed) case



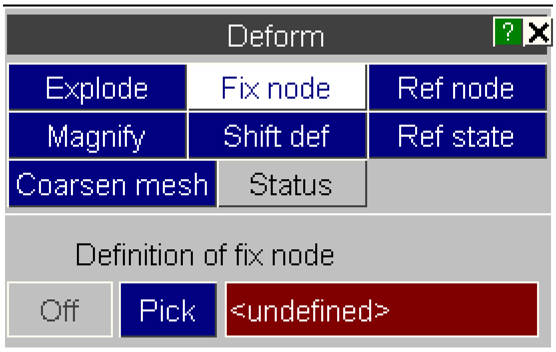
This second figure shows what happens when a node in the dummy's pelvis is fixed using **FIX_NODE.**

Fixed (at pelvis) case



The **FIX_NODE** facility allows you to specify a node that remains fixed at its undeformed position, regardless of any displacements that may occur. This is implemented by finding the displacement vector of the node at each complete state, and subtracting that vector from the coordinates of every node in the model. In the example above one would fix a node on the sled, which would then appear to be fixed in space, and simply see dummy motion within it.

(If a rotation as well as a translation is required you can use **SHIFT_DEFORMED** instead: see SHIFT_DEFORMED Translating and rotating a model back to its undeformed position .)



This figure shows the **FIX_NODE** control panel in its default state: no node is fixed.

To fix a node **PICK** it, or type in its label. Once defined this mode can be switched on or off.

FIX_NODE applies a model space offset to what is drawn: it is a translation of the model, updated for each state.

(It is not the same as **VIEW, CN** (Centre on node) which is a purely graphical transformation that sets the viewing centre for rotations.)

Note 1:	**FIX_NODE** (which applies a translation) and **SHIFT_DEFORMED** (which applies both a rotation and a translation) are mutually exclusive: you can only have one or the other active at one time.
Note 2:	The node used in **FIX_NODE** (which affects the graphical displacements) is the same as that used as a single **REFERENCE_NODE** (which affects the contoured and reported values). They may be used separately or together.
Note 3:	**FIX_NODE** can be used at the same time as "explosion" vectors and magnified displacements: see Note 3 in Notes on explosion vectors for the order of application.

9.4.5. REFERENCE_NODES Calculating Results with Respect to One or Three Nodes

REFERENCE_NODES Calculating Results with Respect to One or Three Nodes

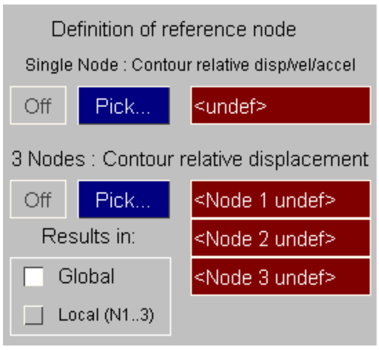
FIX_NODE and **SHIFT_DEFORMED** above affect only how the current image is displayed, they do not change the computed values which are contoured or reported.

REFERENCE_NODES , on the other hand, does not affect the display at all, rather it modified the values that are computed to make them relative to those at the nodes chosen. This feature allows intrusion or relative deformation to be contoured. Two mutually exclusive options are available:

- **Single node** : Displacement, Velocity and Acceleration values are reported relative to that node.
- **Three nodes** : Displacement only is reported relative to node 1, in the coordinate system formed by N1N2N3.

In the Three nodes case results can be reported in either the global or the local (N1N2N3) coordinate system.

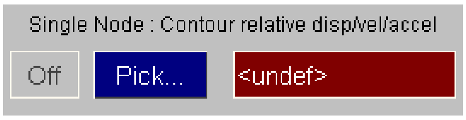
The " **Single** " and " **Three** " node cases are mutually exclusive, you cannot have both active at one time.



Defining one or three nodes

Single Node

You pick a single node <N$_0$>.



Displacements, Velocities and Accelerations are calculated with respect to the value at that node. For example if $\mathbf{V}$ is a velocity vector:

$V'_N = V_N - V_0$

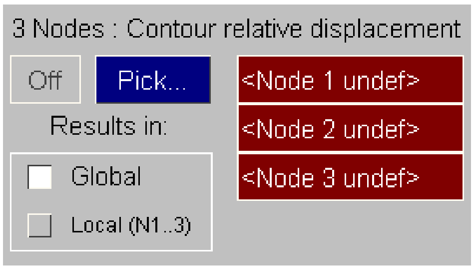
V'_N = modified velocity vector at node <n>

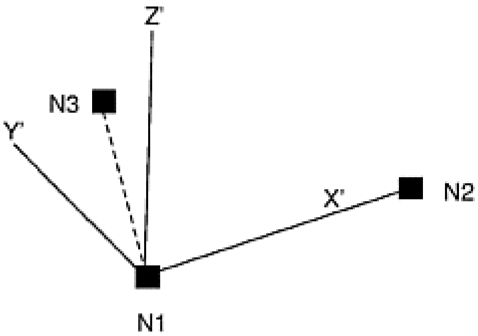
V_N = original velocity vector at node <n>

V_0 = current velocity vector at reference node <N$_0$>.

Three Nodes

You pick three nodes <N$_1$,N$_2$,N$_3$>. N$_1$ is the origin, and the nodes form a right-handed coordinate as for SHIFT_DEFORMED above.





Displacements (only) are calculated with respect to this system such that for displacement vector D:

$D'_N = R \; . \; [D_N - D_0]$

D'_N = modified displacement vector at node <n>

D_N = original displacement vector at node <n>

$\mathbf{D}_0$ = current displacement vector at reference node <N_0>
$\mathbf{R}$ = the rotation matrix to transform back to the selected coordinate system

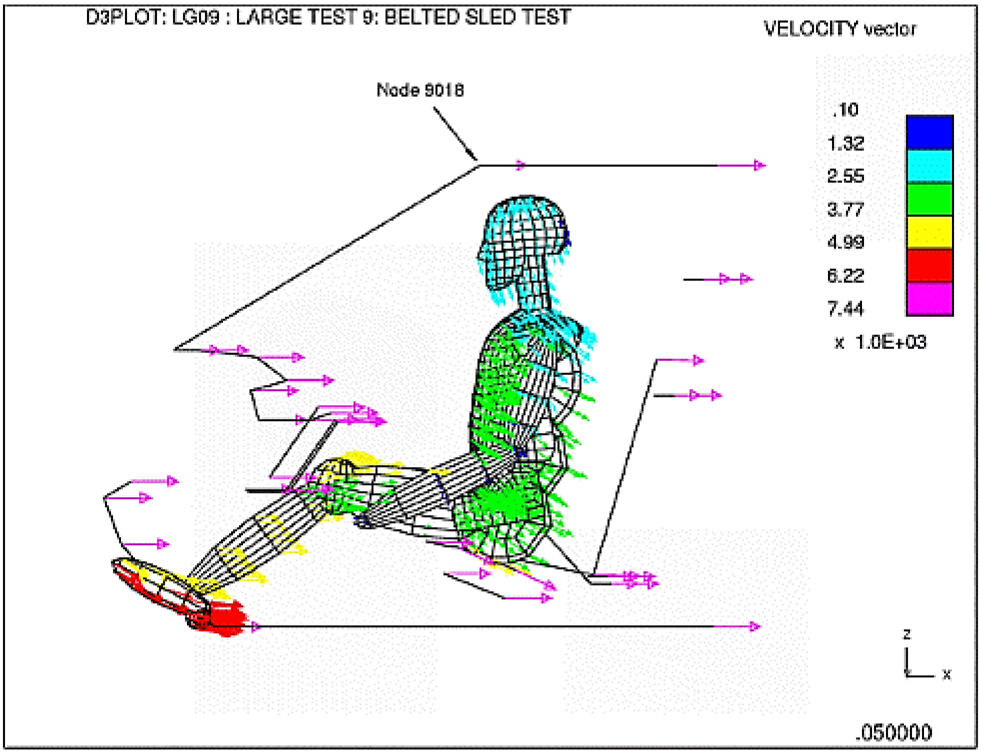
Using REFERENCE_NODE (single node case)

Here is an example showing how a single **REFERENCE_NODE** might be used.

In this case we have a dummy in a sled test, as above, where a crash is simulated by pulling the sled backwards. However what we are interested in is the velocity of the dummy relative to the sled, since in a real crash the sled (= car) would be more or less stationary, while the dummy would still be travelling forwards.

We can achieve this by picking a node on the (rigid) sled as our reference node, and displaying all velocities relative to that.

Here is the "raw" image, showing that the sled is moving rapidly backwards.

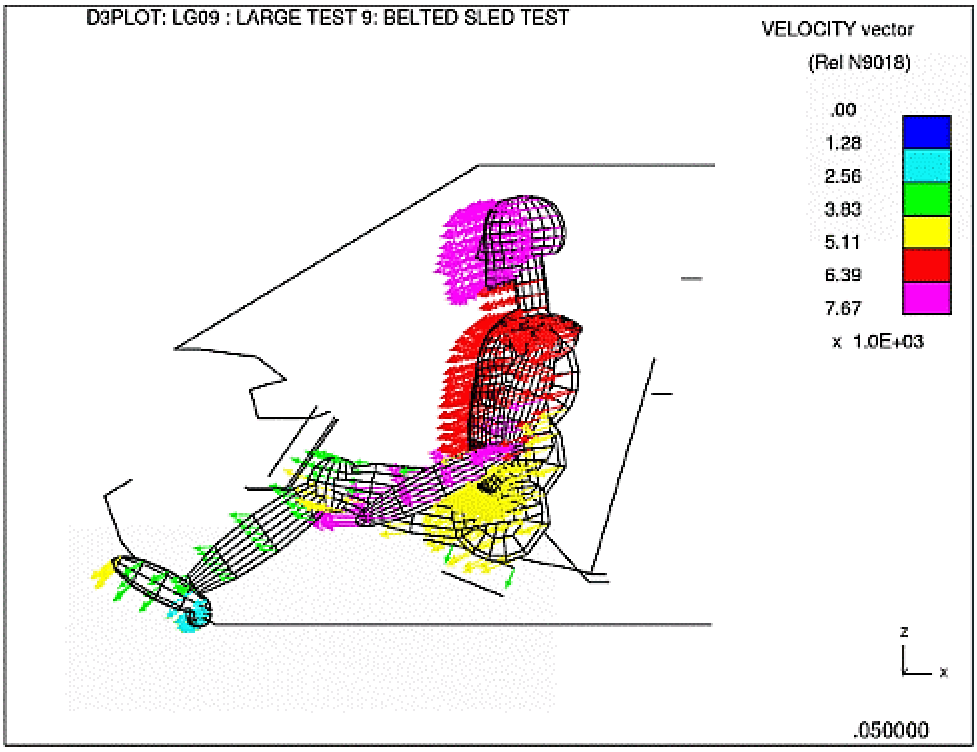


REFERENCE_NODE now switched on .

Here is the revised velocity plot now that the REFERENCE_NODE has been switched on.

The velocities of the sled at node 9018 have been subtracted from all velocities, making those on the dummy effectively relative to the sled.

(Should we wish to fix the sled in model space, and to draw the deformed shape of the dummy relative to that throughout an animation, we could also use FIXED_NODE. However the two operations are independent and do not have to be combined.)



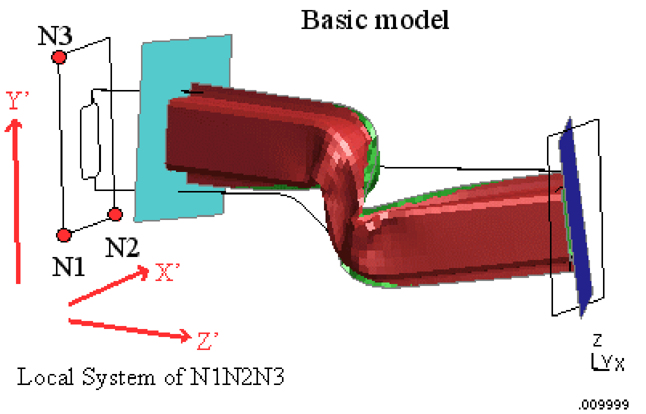
Using REFERENCE_NODES (3 node case)

The following example shows how **REFERENCE_NODES** (3 nodes) works, and how it is related to **SHIFT_DEFORMED.**

Here is the basic model.

It is a crush tube shown in its final state, with the undeformed geometry overlaid.

The loading platens at each end are pushed together, but they are free to rotate. The problem is to determine the maximum "end to end" deformation.



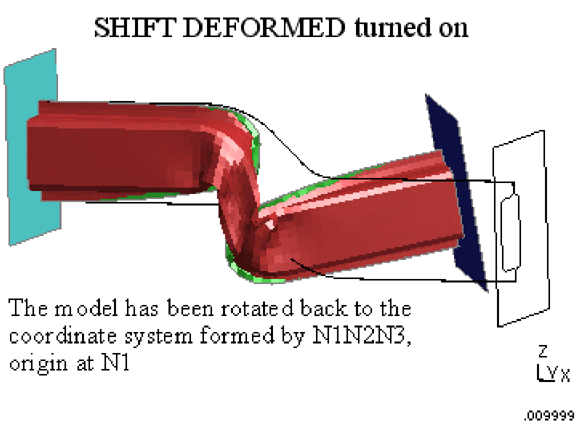
It is clear from this plot that the blue end moves and rotates, and this makes it difficult to determine the deformation relative to that end. The sequence below shows how to overcome this problem.

Three nodes (N1, N2, N3) have been chosen on the loading platen at the blue end, and they form a local coordinate system as shown.

SHIFT_DEFORMED turned on .

This is the same model at the same state, but now SHIFT_DEFORMED has been switched on, and the model has been rotated back to the coordinate system formed by N1N2N3, translated back to origin at N1.

Note that the rotation and translation are back to the undeformed locations of nodes N1 to N3.

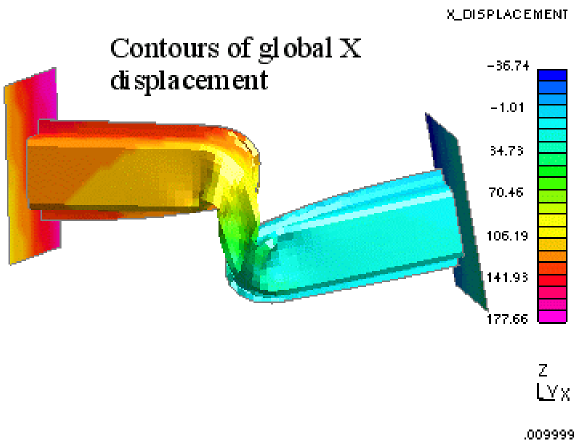


(*This step is not necessary in order to calculate data relative to reference nodes, but it makes the example much clearer.*)

Contours of X displacement now shown.

This plot shows global X displacement, which is approximately along the length of the tube.

However because both ends of the tube have rotated it is difficult to estimate the movement of the two ends relative to one another. We can see that it approximately 177.66 + 36.74 = 214.4, but this may not be good enough.

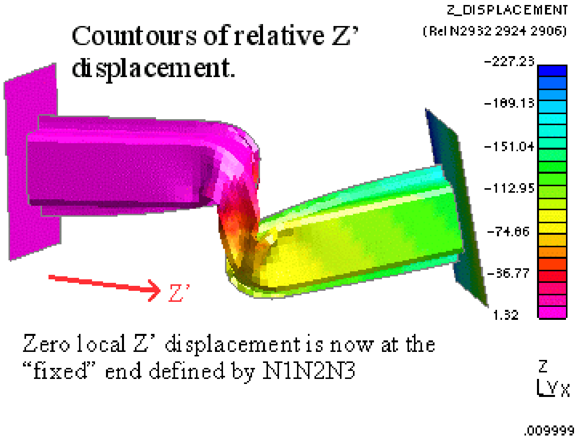


In order to obtain a more accurate value it is necessary to express the displacements in terms of the coordinate system formed by N1N2N3.

Remember: SHIFT_DEFORMED ***only*** affects the deformations drawn, it has no effect on the values that are contoured or written out.

REFERENCE_NODES turned on, and contours of local Z' displacement shown.

By switching on REFERENCE_NODES, and selecting output in the local system, we can now plot displacements in the local Z' direction relative to the left hand end.

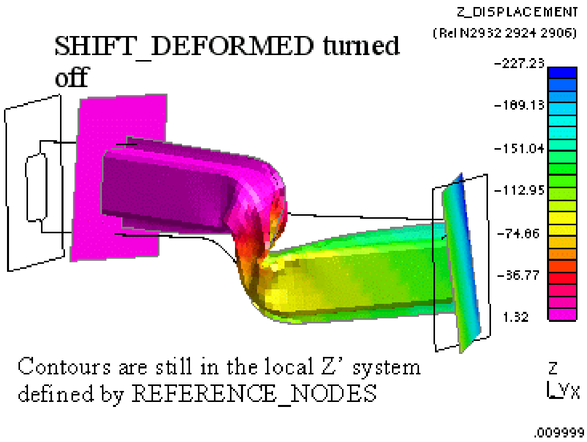


It is now clear that the actual peak movement at end two is actually 227.23, somewhat higher than our estimate from the approximate global X plot above.

This technique is very useful when calculating "knock-back" and "intrusion" displacements at particular locations in a model.

SHIFT_DEFORMED turned off, but REFERENCE_NODES left on.

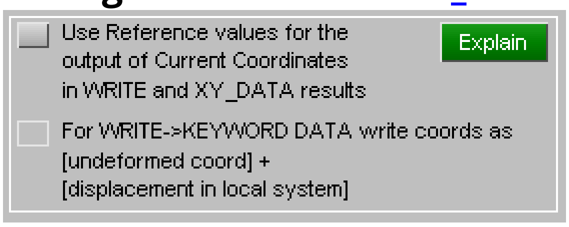
This plot demonstrates that while SHIFT_DEFORMED and REFERENCE_NODES are related, and share the same nodes, they can act independently.



SHIFT_DEFORMED has been turned off, so the deformed shape is now the "true" shape, but the contours are still expressed in the local Z' of the axis system defined by N1N2N3.

This is a harder plot to understand, because the axis system of the plotted results is not that easy to discern.

Reference node settings in WRITE and XY_DATA output



By default the scalar output of nodal coordinates in WRITE and XY_DATA will not take into account any reference node values, but selecting the option to use reference values causes them to be considered, giving numerical values equivalent to those that appear in the plots.

- For a single node coordinates will be in the global cartesian system with the coordinates of node N1 subtracted, ie the effective origin [0,0,0] is at the coordinates of node N1 at the reference state.
- For three nodes the coordinates will be reported in the local system N1N2N3, with the effective origin [0,0,0] offset to coordinates of N1 at the reference state.

Additionally, there is an option to WRITE coordinates as [undeformed] + [displacement in local system].

WARNING :

> Since reference nodes can be defined on a "per-window" basis, but WRITE and XY_DATA are "per-model", there is a potential ambiguity if multiple windows on a model have been defined as having different reference nodes - which window's settings will be used for the written/graphical output?
>
> The answer is those of the most recently drawn window, which is not easy to determine reliably. Therefore if you are planning to use this option you are strongly advised:

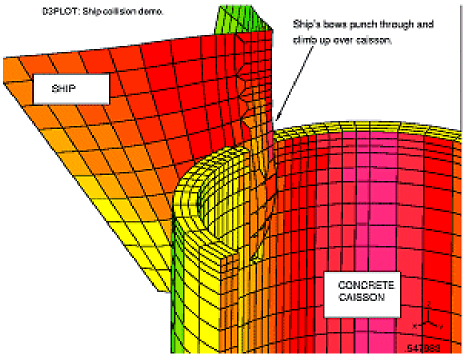
either :	Only to have a single window open on the model
or :	If you have multiple windows open, to ensure that all of them have the same reference node settings.

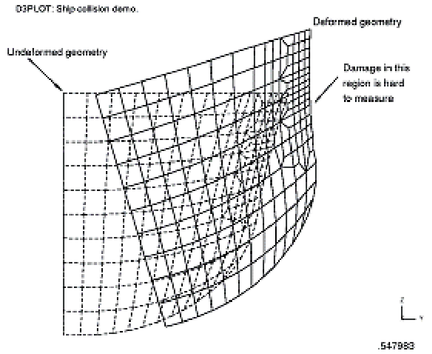
9.4.6. SHIFT_DEFORMED Translating and Rotating a Model Back to its Undeformed Position

SHIFT_DEFORMED Translating and Rotating a Model Back to its Undeformed Position

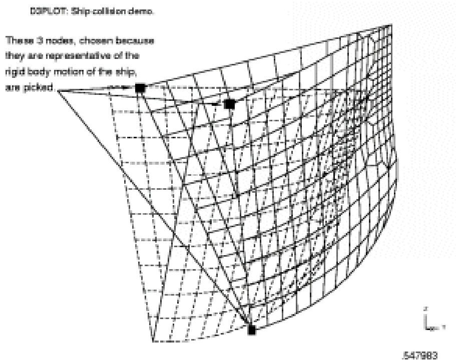
Sometimes it is useful to be able to move a deformed structure back to its undeformed position, for example to measure knock-back (crush) following an impact. In many cases this will involve applying a rotation as well as a translation, and the **FIX_NODE** option described in FIX_NODE Fixing a node position despite displacements (which only applies a translation) will not be adequate.

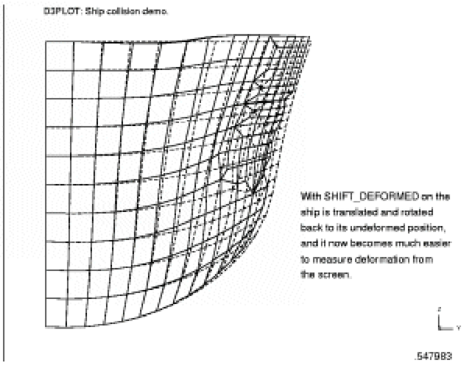
Consider the following example: a ship hits a concrete caisson, punches a hole through it with some damage to its bow plates, and also pitches up as at tries to climb over the caisson. Measure the damage to the bow plates. The situation is shown in the figure below left, and in the figure below right the deformed and undeformed shapes of the ship are shown. Clearly the rotation the ship has undergone makes it hard to measure the deformation





By using **SHIFT_DEFORMED** to pick three nodes that are representative of the rigid body motion of the ship, translation and rotation can be applied to bring the deformed geometry back to overlay the undeformed, making measurement possible: see the figures left and right below).

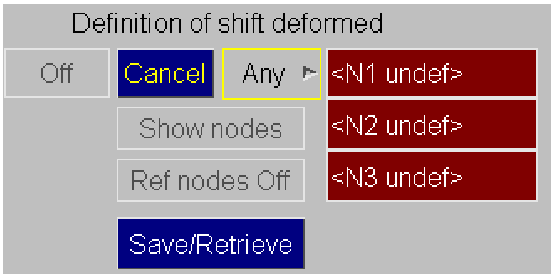




The **SHIFT_DEFORMED** panel is shown right.

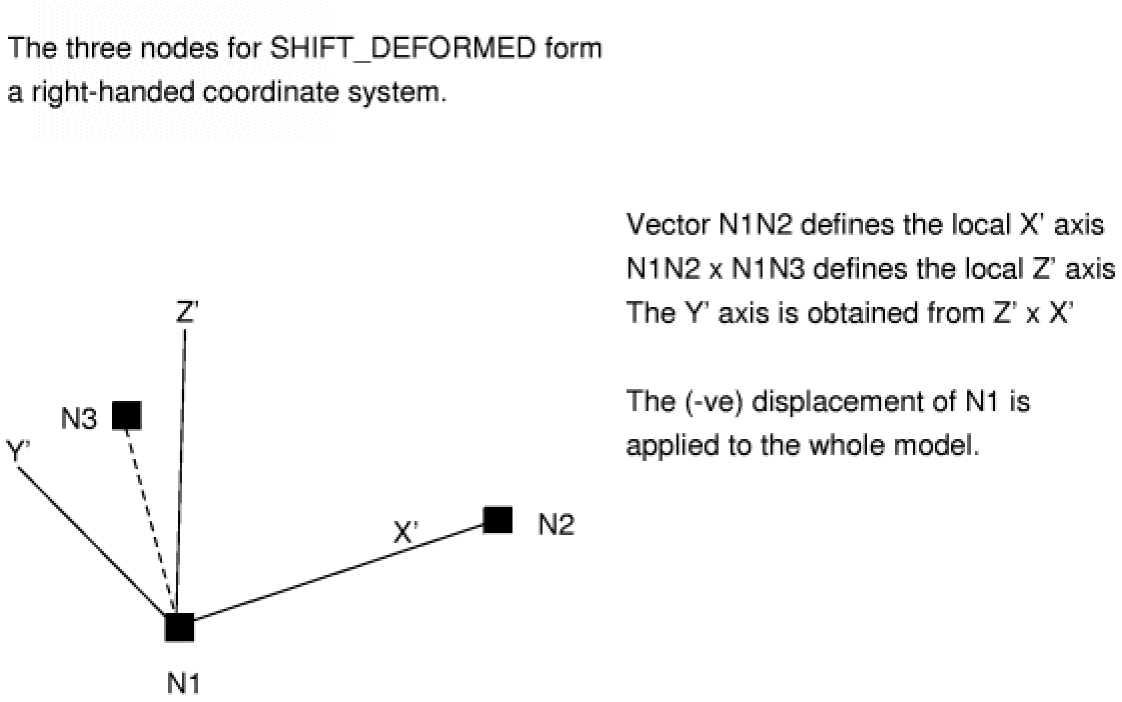
You can screen- **PICK** the nodes, or type them in directly.

Once defined this mode can be switched on or off at will.



Choosing sensible nodes for SHIFT_DEFORMED .

The three nodes you choose form a right handed coordinate system, so they must not be colinear (or become colinear due to displacements), and the order of their definition is significant: see the figure below.



The local coordinate system [X',Y',Z'] is defined as shown here, and the inverse of this is applied to rotate the model back to its undeformed state.

The displacement of node 1 is subtracted from all nodes in the model to bring it back to the undeformed position.

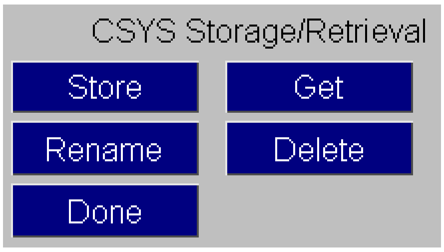
(**FIX_NODE** applies this translation only, thus it is a subset of **SHIFT_DEFORMED** , which is why the two operations cannot be used at the same time.) You should try to choose three nodes whose relative position will not change too much as the model deforms, so that their motion is representative of the rigid body motion of the structure as a whole. And node 1, from which the rigid body translation if computed, is the most significant. For example in a frontal impact car crash analysis you should nodes at the back of the car. If your model has some rigid bodies then nodes on them would be ideal.

Note 1: **SHIFT_DEFORMED** cannot be used at the same time as **FIX_NODE** , since the translations they apply would conflict.

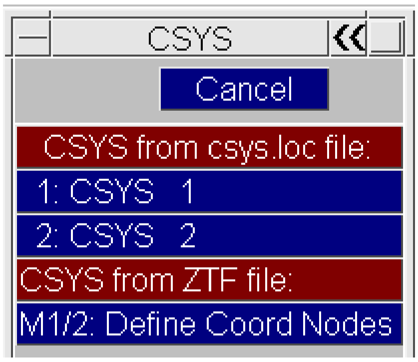
Note 2: **SHIFT_DEFORMED** can be used in conjunction with "explosion" vectors and magnified displacements. The order of application is given in Note 3 in Notes on explosion vectors.

Note 3: **SHIFT_DEFORMED** uses the same three nodes as those in **REFERENCE_GEOMETRY** . The difference is that shifting the model simply changes the graphics displayed, whereas reference geometry changes the data values contoured and output. They can be used in conjunction or separately.

The **SAVE/RETRIEVE** button allows multiple local coordinate system definitions to be saved, retrieved and deleted from a 'csys.loc' file, written in the model directory. This means that local coordinate systems can be reloaded across different sessions of D3PLOT without having to recreate them.



Pressing the **GET** button brings up a list of available coordinate systems in both the 'csys.loc' file and any *DEFINE_COORDINATE_NODES definitions in the ZTF file.



9.4.7. REFERENCE STATE/MODEL

REFERENCE STATE/MODEL

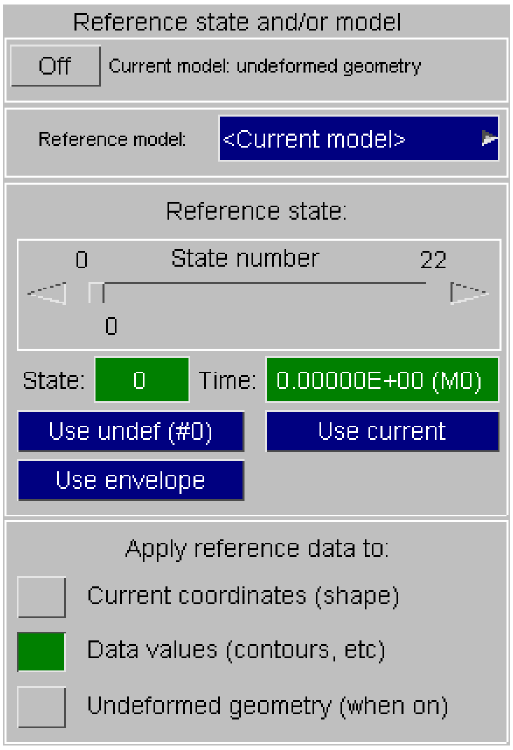
Normally results at a given state are drawn and reported verbatim, subject to the various options above. However it is possible to subtract from the current data:

- The results at a different state in this model, showing the difference between two times.
- The results at the same, or a different, state in another model, showing the difference between models.
- If all active models have been envelope plotted the **USE ENVELOPE** button will become active and you will be able to do a relative plot of the envelope plots between models.

This operation is applied to all nodal and element data, and its application can be to any permutation of:

- The current graphical coordinates (ie the plotted shape)
- The current data values (ie contoured, written and time-history values)
- The current undeformed geometry plot.

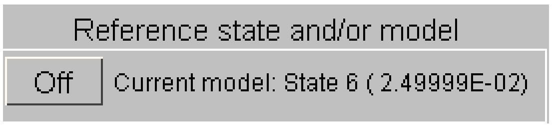
By default the reference model is the current model, and the reference state is zero, and "reference plotting" for a window is turned off, meaning that no action is taken here.



9.4.7.1. Turning Reference Plotting On/Off

Turning Reference Plotting On/Off

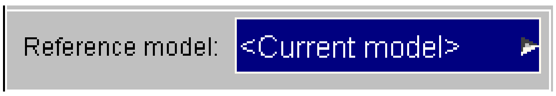
Reference plotting is not active in any window until it is turned on in that window.



Settings for each window are stored separately, and this panel shows those for the first active window selected by the **`W1 .. Wn`** tabs. To define different settings in different windows select a single **`Wn`** tab at a time, and configure each window individually.

9.4.7.2. Choosing the Reference Model

Choosing the Reference Model



Two possibilities exist:

Plot relative to the current model

This is the default case. You select a reference state in the current model, and results are plotted relative to this state. If you have multiple models in a window then each model will be plotted relative to itself.

Data from the reference state are subtracted from those in the current state, and the results displayed. It is perfectly possible to choose a reference state later than the current one, and hence to get "negative" results, the computation is simply:

<data> $_{\text{displayed}}$ **= <data>** $_{\text{current}}$ **- <data>** $_{\text{reference}}$

Plot relative to another model

This is the more complex case of plotting data from this model relative to a state from a different model. If you have multiple models in a window each model in that window will be plotted relative to the reference model.

The principle is exactly the same: the reference data is subtracted from the current, but mapping of reference model onto current is done as follows:

- **Mapping is "by external label".**

  The results from node label <i> in the reference are subtracted from those for node label <i> in the current model. And likewise for elements.
- **No checking for geometrical or topological proximity takes place.**

  No check is made that node <i> (or element <j>) in the two models are equivalent, or even remotely in the same place - either topologically or geometrically.
- **If no equivalent label is found, zero is reported.**

  If no matching node or element can be found in the reference model, then zero is reported as a result - regardless of the actual value in the current model.

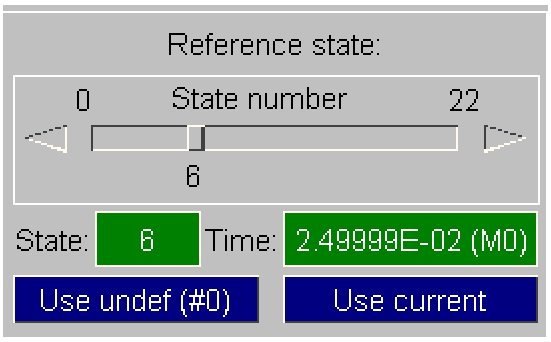
This means that models which are topologically nearly identical compare well, but areas which have been remeshed may give very misleading comparisons.

9.4.7.3. Choosing the Reference State

Choosing the Reference State

Whether you are using the current or a reference model you must define which state in that model is to be used as the "reference" one.

You can use either a fixed state number, or a changing "current" one.



Using a fixed reference state

This is the default case, invariably used when plotting data relative to the current model.

You simply select a valid state from the model, and this is used as the reference state. It is perfectly legal to select a fixed state from a different model too.

Using the "current" state

This meaningless if the reference model is the current model, since results will always be zero.

But if plotting relative to a different reference model it is a powerful tool, especially during animations:

```
For each state <i> the data from the equivalent state
<i> in the reference model is subtracted.
```

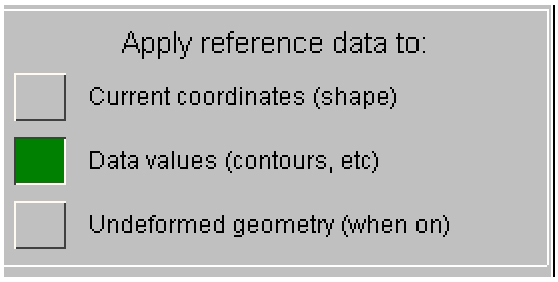
This means that you can see how the differences between two models vary over time. Obviously it is important to make sure that the states in the two models have the same times, as no check is made for time equivalence.

9.4.7.4. Choosing what Reference Data Applies to

Choosing What Reference Data Applies To

You can control the extent to which "reference" logic applies to plotted geometry and contoured or written data.

Any permutation of the below can be selected.



Current Coordinates	Whether or not reference geometry is used for the shape that is plotted on the screen. Note that "coordinates used for plotting" and the component "displacements" are kept separate, and can be controlled individually.
Data Values	These are the values contoured, vector plotted, written by the WRITE command and reported by XY_DATA.
Undeformed Geometry	This is the display of undeformed geometry on the plot only (when drawn). Normally this will always display state #0, but if you turn this on the geometry of the reference state will be used instead.

9.4.7.5. Examples of Using Reference State/Model

Examples of Using Reference State/Model

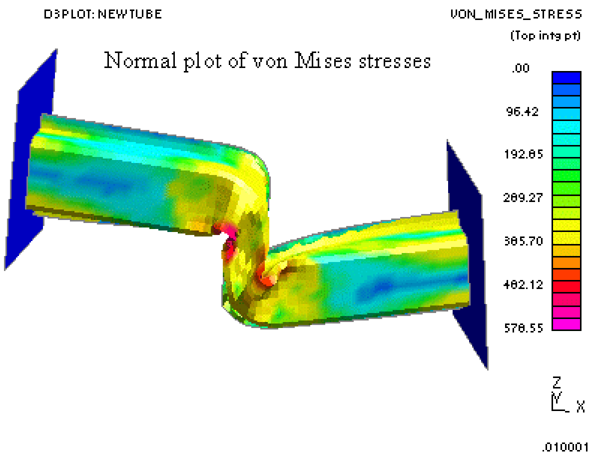
The following examples demonstrate how the feature might be used.

Example 1: Reference state in the same model

Here is the "raw" plot - Von Mises stress in a crush tube.

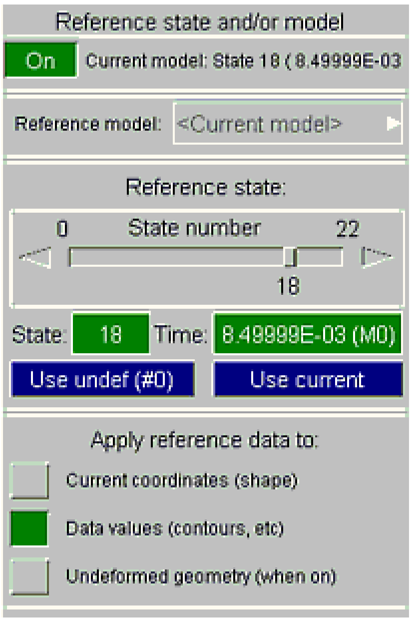
And below is the same plot with the reference state set to #18, which is at t = 0.0085s, in the same model. Remember that:

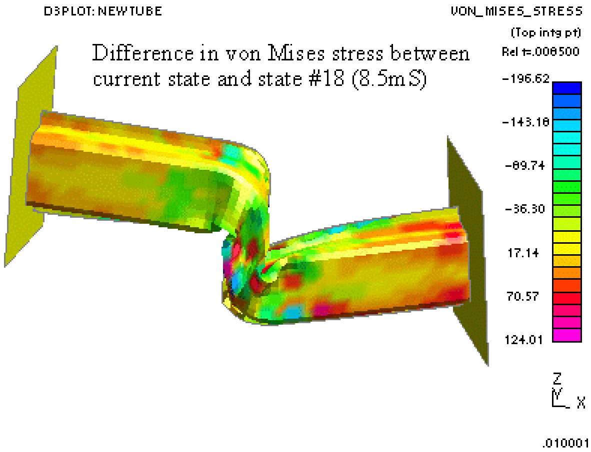
<data> displayed = **<data>** current - **<data>** reference



Therefore, as in this example, negative von Mises stresses can be generated since, in some locations, the stress at an earlier state was in fact greater.

(Incidentally this illustrates that the reference calculation is performed upon the scalar output of the data component computation, not upon its individual tensor components.)



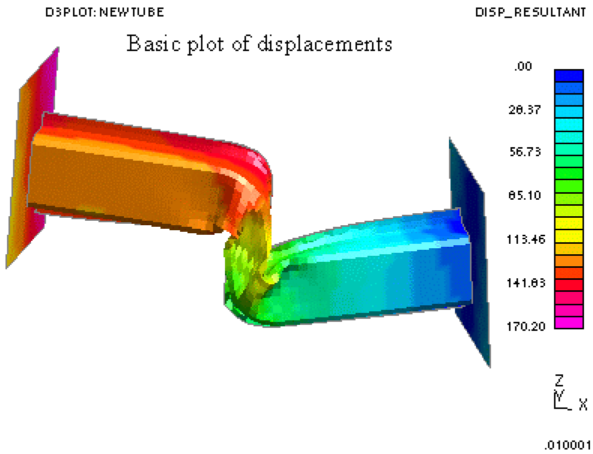


Example 2: Reference state in a different model

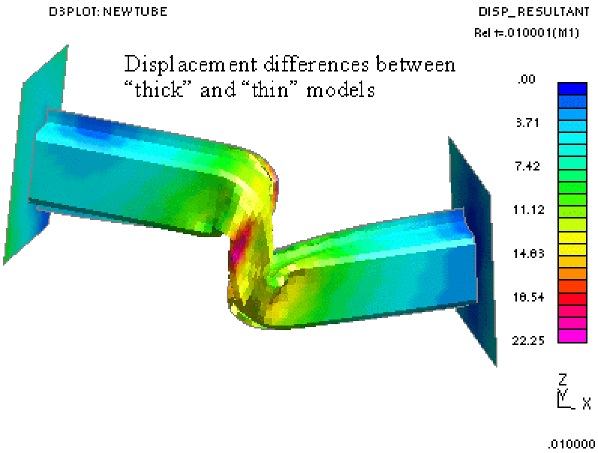
In this case the model above has been re-run, but with the section thickness of the crushable elements reduced by 25%. Here we are comparing the results between original and modified models to see what difference this makes.

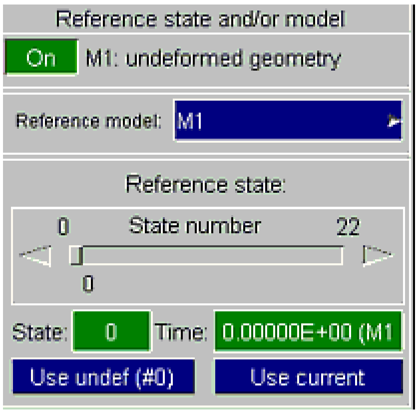
Obviously rerunning the same model with different section properties does not upset node or element labelling, so exact equivalence between the two analyses is preserved.

Here is the original image, showing contours of "true" displacement magnitude.



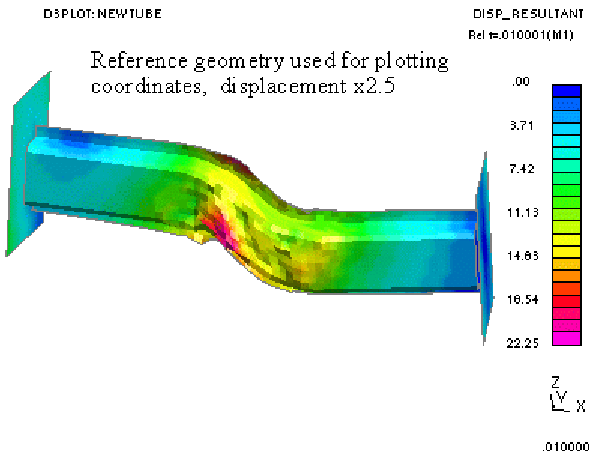
Here is the same image, with the displacements in the reference model subtracted from the "true" ones above. The "current" state has been used in the reference model (M1), and this is reported on the plot.

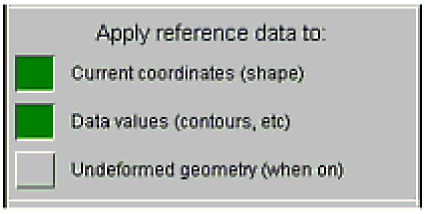




It is not that easy to visualise the differences in shape from the contour plot above, since they are obscured by the gross deformations of the structure.

So in this plot reference geometry has been used for the current (plotting) coordinates as well, showing the difference in displacement between the two models. Effectively this is the undeformed geometry + the difference in displacement between the two models.





Displacements have also been magnified by a factor of 2.5 using **MAGNIFY_ DISPLACEMENTS** to exaggerate them, making them clearer.

REFERENCE STATE - Notes

1) If the reference state is set to state 0 then all values reported will be the absolute values.
2) If an analysis contains pre-stressed elements then state 1, not state 0, should be selected if values relative to the pre-stressed values are required. (State 1 is a genuine set of results at analysis time zero, state #0 is a synthesised set of zero values.)
3) The reference state option is not available for a model that includes adaptivity, see Support for analyses using "Adaptive Remeshing".
4) Using the reference state option will increase the amount of memory used by D3PLOT slightly, as two complete states have to be stored simultaneously.

REFERENCE MODEL - Notes

1) Any model can be used as a reference model, but it should be reasonably similar to the original if sensible results are to be obtained.
2) Using a reference model will slow down plotting since the <current> vs <reference> lookup by label imposes an overhead. It can also slightly increase memory usage.

9.4.8. TRANSFORM

TRANSFORM

Applying translation, rotation, reflection and scale to a model as it is read in.

By default model data are read in verbatim, but it is possible to apply transformations to them as they are read so that the data stored in memory is "as transformed". This can be useful if you wish to overlay models, or perhaps to compare left and right-handed versions of the same model.

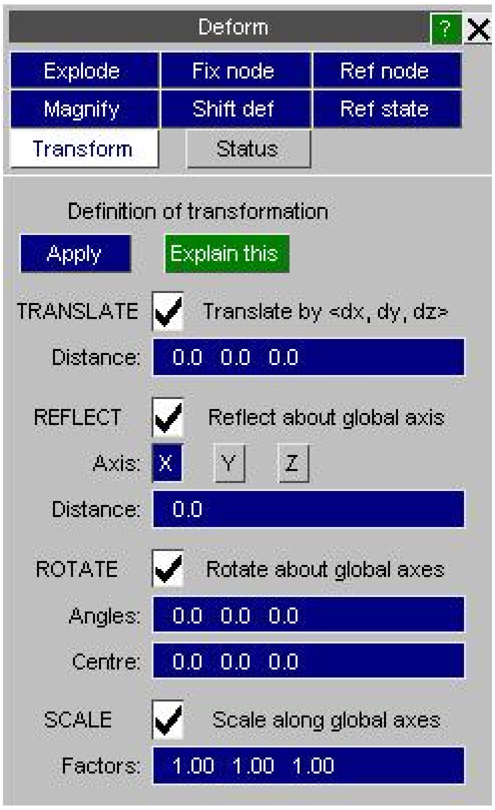
Any combination of the following may be applied:

TRANSLATE	Apply a [dx,dy,dz] translation
REFLECT	Reflect about a point on the X, Y or Z axes
ROTATE	Rotate by angles [theta x, theta y, theta z] about an [x,y,z] centre of rotation
SCALE	Apply scale factors [sx,sy,sz]

All transformations are applied in the global axis system in model space, and if multiple transformations are specified they are performed sequentially in the order above.

Each transformation must be turned on using its [tick] box in order to be active, its parameters must be defined, and finally **Apply** must be used to apply the current transformation(s) to the model.

Transformations apply to the specified model only, and may be applied, modified or cancelled at any time. Each such change results in all data currently in memory being deleted and reread as required in its new form.



9.4.8.1. TRANSLATE

TRANSLATE

Translate model by vector [dx,dy,dz]

By default no translation is applied, but you may apply a vector [dx,dy,dz] in global model space.

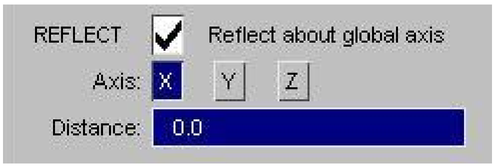
Translation is applied to:

Coordinates	**`(New coord) = (old coord + translation vector)`**

9.4.8.2. REFLECT

REFLECT

Reflect model in one of the global X, Y or Z axes about a point on the relevant axis



Firstly select which of the X, Y or Z axes to reflect about, and then the distance from zero (ie position) on that axis where the reflection plane is to be located.

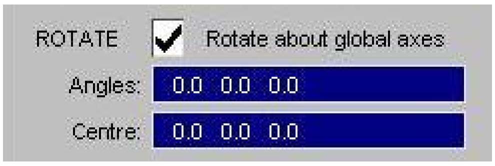
Reflection is applied to:

Coordinates	For the relevant coordinate **`c`** *new* = **`Distance - (c`** *old* **`- Distance)`**
Velocity and acceleration vectors **Other vector data** (eg forces/moments ex binout file)	The sign of the relevant term is changed
Stress and strain tensors **Other tensor data** (eg ex binout file)	The sign of off-diagonal (shear) terms with the reflection axis in is changed. For example reflection about the **X** axis results in T **x** y and Tz **x** terms changing sign.

9.4.8.3. ROTATE

ROTATE

Rotate model by angles [Tx,Ty,Tz] about centre of rotation [Cx,Cy,Cz]



Define rotation angles, which are specified in degrees about the global X Y Z axes, and also the centre of rotation.

Rotation is applied to: (`[R]` is the 3x3 rotation matrix, `[R']` is its transpose, "New" and "Old" below are vectors or tensors as appropriate)

Coordinates	`New = [R] x [Old - centre] + centre`
Velocity and acceleration vectors **Other vector data** (eg forces/moments ex binout file)	`New = [R] x Old`
Stress and strain tensors **Other tensor data** (eg ex binout file)	`New = [R'] x Old x [R]`

Compound rotations about more than one axis

If rotation angles about more than one axis have been specified they are applied in the order Tx, then Ty, then Tz; in other words `[R] = [Rz].[Ry].[Rx]` . If you need to apply compound rotations about more than one axis it is recommended that you check the outcome carefully to make sure that you have achieved what you intended.

If you already have a set of direction cosines and you want to know the angles required to reproduce these they can be computed as follows:

[Rx]: Rotation about the X axis:	**[Ry]: Rotation about the Y axis:**	**[Rz]: Rotation about the Z axis:**
`Sx` = Sin(theta X) `Cx` = Cos(theta X)	`Sy` = Sin(theta Y) `Cy` = Cos(theta Y)	`Sz` = Sin(theta Z) `Cz` = Cos(theta Z)

`[ 1 0 0 ]`
`[ 0 Cx -Sx ]`
`[ 0 Sx Cx ]`
	`[ Cy 0 Sy ]`
	`[ 0 1 0 ]`
	`[ -Sy 0 Cy ]`
		`[ Cz -Sz 0 ]`
		`[ Sz Cz 0 ]`
		`[ 0 0 1 ]`

Concatenating these together in the order [X, Y, Z], ie `[Rz].[Ry].[Rx]` gives the compound rotation matrix of cosines `[Rc]` :

```
[  Cy.Cz    Sx.Sy.Cz - Cx.Sz    Cx.Sy.Cz + Sz.Sz  ]
[  Cy.Sz    Sx.Sy.Sz + Cx.Cz    Cx.Sy.Sz - Sx.Cz  ]
```

```
[   -Sy          Sx.Cy               Cx.Cy        ]
```

From which it can be seen that a set of Euler angles can be extracted as follows (using the notation < `ij` > is row < `i` >, column < `j` >)

`Theta X = arctan(32/33)` Since (Sx.Cy / Cx.Cy) = (Sx / Cx)
`Theta Y = arcsin(-31)`
`Theta Z = arctan(21/11)` Since (Cy.Sz / Cy.Cz) = (Sz / Cz)

Therefore if you have 3x3 matrix **`[Rc]`** simply calculate the theta angles using the equations above, convert to degrees and apply as [Tx,Ty,Tz] to get the same effect in D3PLOT.

9.4.8.4. SCALE

SCALE

Scale model by factors [Fx,Fy,Fz] in global axes

Define factors in each of the global X Y Z axes.

Scale is applied to:

Coordinates **`New = Old x Factor`**

9.4.8.5. Apply

Apply

Applies the currently defined transformation(s)

The following actions take place:

- All data that has been read in for the active model are deleted from memory
- All windows that reference this model have their cached data deleted, and also any element normals (for lighting) that have been computed are deleted.

This means that any future operations referencing the active model (plots, WRITE, etc) force a fresh "read from disk" operation during which the new transformations are applied.

For large models on a slow or remote disk this can be a slow operation (it is nearly equivalent to closing and reopening the model), so it is best to get transformations sorted out ***before*** building large animations.

9.4.8.6. The Order of Multiple Transformations

The Order of Multiple Transformations

Where more than one transformation is active they are applied in the order they appear above, i.e.:

1. Any translation
2. Any reflection
3. Any rotation
4. Any scale

Therefore when specifying multple translations you may need to consider how an earlier operation affects a later one. For example if you translate and reflect then the distance along the axis at which you reflect will need to take into consideration any prior translation down that axis.

9.4.8.7. Some Data are Not Transformed

Some Data are Not Transformed

At present the following items are not affected by transformations:

User-defined data	Simple formulae or JavaScripts operating on internal data are already working on "as transformed" data, so it would be wrong to apply further transformations. User data read from external Ascii files are also not transformed. It is not possible to tell whether vector data from these files are spatial coordinates or other directional vector data, and while it would be possible to transform tensors it is more consistent to take the view that all transformations of external data must be applied externally. For similar reasons UBIN components created in the JavaScript interface are also not transformed.
External "blob plot" data	This is always used verbatim.

9.4.8.8. Saving and Reloading Transform Data

Saving and Reloading TRANSFORM data

Transformation data is a "per model" attribute and, as such, it is written to the properties file if this is saved, and hence reloaded when the properties file is reread on input or subsequently by direct command.

This also means that transforms will be "remembered" during Oasys Ltd. sessions.

9.4.8.9. Command-Line Syntax

Command-Line Syntax

For batch usage it may be more convenient to specify transformations using the command line. The operations above are under **/DEFORM, TRANSFORM** and are organised as follows:

---+--- **TRANSLATE**	**Tx Ty Tz** or **OFF**	Translate by Tx Ty Tz, eg: **translate 10.0 0.0 -100.0** **translate off**
+--- REFLECT	**Axis Distance** or **OFF**	"Axis" is **X** or **Y** or **Z** , "distance" is position on axis, eg: **reflect Y -1500.0** **reflect off**
+--- ROTATE	**Tx Ty Tz Cx Cy Cz** or **OFF**	Tx Ty Tz are rotation angles in degrees, Cx Cy Cz is centre of rotation, eg: **rotate 0 0 30 100.0 10.0 -20.0** **rotate off**
+--- SCALE	**Sx Sy Sz** or **OFF**	Scaling by factors Sx Sy Sz, eg: **scale 2.0 2.0 2.0** **scale off**
+--- CANCEL	<No arguments>	Turns off ***ALL*** transformations (leaving values unchanged)

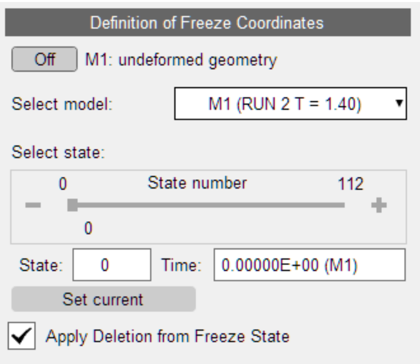
There is no "Apply" command in command-line syntax, as transformations are active as soon as they are specified.

9.4.9. FREEZE_COORDINATES Displaying Results On Fixed Geometry

FREEZE_COORDINATES Displaying Results on Fixed Geometry

This option can be used to display the results from any state on top of the deformed geometry.

The **FREEZE_COORDINATES** control panel is shown in the figure (right).

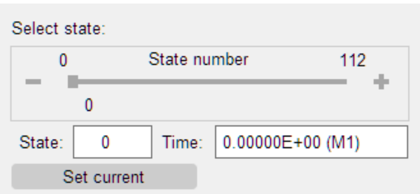


This option can be applied separately to each combination of Model and Window. If the currently active Windows contain multiple Models then the **Select Model** dropdown can be used to select the model that you want to change the settings for. If the selected model is in more than one of the currently active windows then any settings you change are applied to all the active windows the model is in.

This option is similar to using **MAGNIFY_DISPLACEMENTS** with the magnification set to 0.0 to display results on a fixed geometry but unlike **MAGNIFY_DISPLACEMENTS** this option can be used to fix the geometry at any state within the analysis.

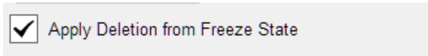
Any changes take effect the next time you update the plot, and stay in effect until changed again.

The state used by this option can be set by either the state slider, entering a state number or by entering a time value. When a time is entered the state nearest to the specified time is selected.



Set Current can be used to set the Freeze State for each model/window combination to the state that is currently being displayed in the window.

By default element deletion from the Freeze State is used when display the results from other states on the Freeze State geometry. This option can be turned off so that the element deletion from the state the results are from is used but this can lead to plots where elements are heavily distorted where nodes have moved large amounts where they have become free when the elements attached to them have been deleted.



9.5. CUT SECTIONS

Cut Sections

The **Cut Section** menu is invoked from the Tools menu or from keyboard shortcut **x**.

A cut-section, sometimes referred to as a "cutting plane", is a flat plane that cuts through the model. It may be located anywhere in space and oriented at any angle.

When the **Cutting switch** is turned on the intersection of the plane with the model is calculated and the interpolated cut plane is drawn.

This is possible in all D3PLOT display modes, (including animation), and for those that display data this will be displayed on the cut plane.

Various options, described below, define if and/or how the model either side of the plane is drawn.

The forces acting on the cut-plane, integrated from element stresses, may be calculated and output.

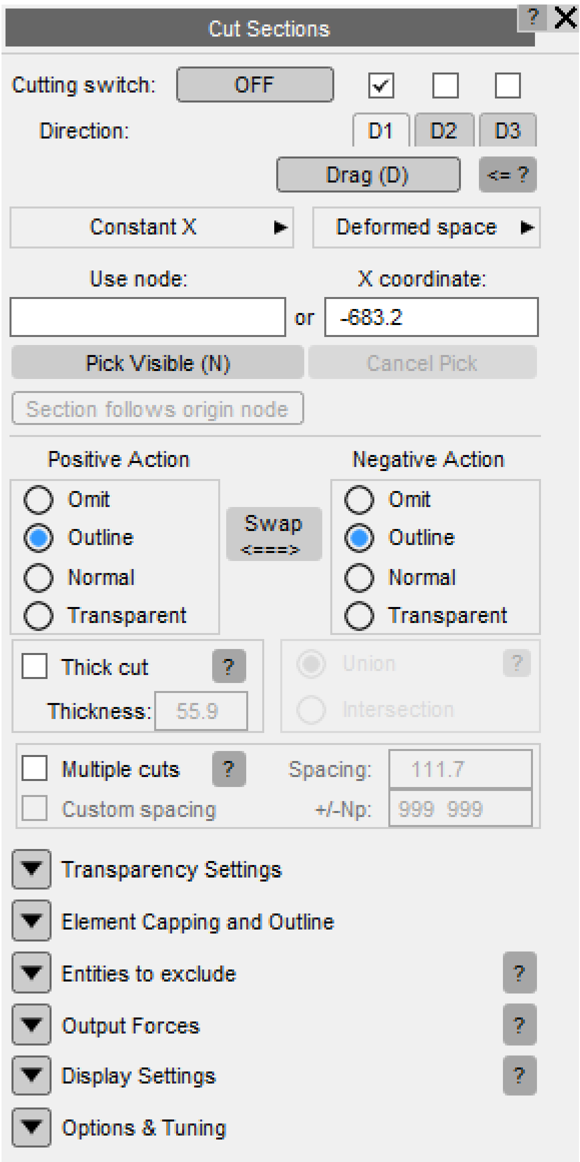
9.5.1. Cut Sections are a "Per Window" Attribute

Cut Sections Are a "Per Window" Attribute

Cut-section definitions apply to all those windows which have their `W1 .. Wn` tabs set. They are stored as an origin coordinate and a local coordinate system, which cuts through all models in the relevant windows.

If you use the **PICK NODE ...** options to derive a coordinate from a node you will be forced to define which model to pick from, but thereafter the coordinate is model-independent.

If you use the option to track node motion across multiple models then special rules apply: see **Section follows nodes**.



9.5.2. Some Important Rules Governing Cut Sections that Must be Clearly Understood

Some important rules governing cut sections that must be clearly understood:

- Cut planes can be defined in up to three directions potentially with multiple parallel planes. Planes may be stored on disk and retrieved at will.
- The plane(s) will only be active when the cutting switch is on and the checkbox for the corresponding direction D1, D2 or D3 is selected. The cutting switch can be used to turn off all cut planes and to turn them back on later. The switches for D1 etc. turn off or back on the plane in the corresponding direction, possibly together with parallel cuts. By default only direction D1 is selected, but the main

cutting switch is off, so a single plane appears as soon as the cutting switch is turned on.

- Only Solid, Shell, Beam and Thick shell elements are cut. Other element types, such as joints, springs, stonewalls, etc, are unaffected. You may want to remove these from the display when using cut planes since they will span the plane.

- Forces and moments on cut planes are also only calculated for elements of these four types which are unblanked, the others are ignored. This is because these are the only element types for which stress &/or force results are consistently available.

- Forces and moments are calculated from solid and thick shell stresses, shell force and moment resultants, and beam forces/moments. Therefore if any of these are rigid no forces will be computed for the relevant materials, even though the elements may be carrying load.

- **Forces** on planes are calculated reasonably accurately. **Moments** are only approximate and should only be treated as (usually under-) estimates.

9.5.3. Creating a Cutting Plane

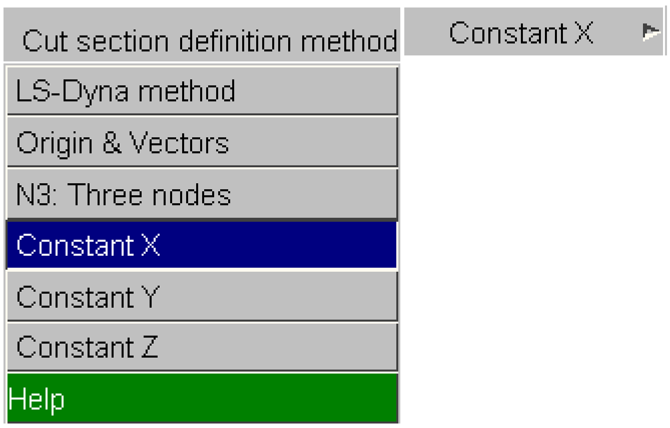
Creating a cutting plane

The first step in creating a cutting plane is to choose how you are going to define it.

A plane is defined by its origin and its local X', Y' and Z' vectors.

The top two options permit the section to be arbitrarily oriented in space, the lower three align it exactly with the model X, Y and Z axes respectively.

Regardless of how it is defined initially the internal definition of the plane is the same, and it may be translated and rotated at will later.



LS-DYNA Method

This option allows import of definitions in the format used by the LS-DYNA *DATABASE_CROSS_SECTION keyword:

- Normal vector tail coordinate
- Normal vector head coordinate
- Edge vector head coordinate

If you have written a .ZTF file from PRIMER than any database cross sections in your original input deck can be imported by using the **Import DATABASE_SECTION** option.

Note: LS-DYNA cross sections use lagrangian ("basic" in D3PLOT terminology) space.

When you define a cut section using this method you will be asked if you want to swap to "basic" space for compatibility with LS-DYNA. This is explained in more detail below in Defining a space system for the plane . (See Appendix B for an oa_pref option that will allow you to set this as your default definition method.)

Origin and Vectors Method

This definition requires the user to enter (in model coordinate space) the:

- coordinates of the origin for the plane
- the local x-axis vector
- any vector lying in the local XY plane.

N3: Three nodes method

This method requires you to pick three nodes which form the local axis system as follows:

- Node 1 is the origin
- Node 2 gives the local X axis from the vector |N1N2|.
- Node 3 gives the local X axis from the vector |N1N2|.

Normally the coordinates of the nodes at the current state form the basis of the plane definition, but if you choose " Section follows nodes " you can update the plane at every state as the nodes move.

Constant X,Y,Z Method

The three "constant" values allow you to define the coordinate along the model X, Y or Z axes respectively at which a plane of that constant axis value will be defined. Locally:

- The origin will be [0,0] on the other two axes.
- Local Z is in the +ve direction down the axis chosen.
- The other two axes are chosen for you, aligned with the two unchosen model axes.

If you use a node here then normally the coordinates of that node at the current state are used, but if you choose " Section follows nodes " you can update the plane at every state as the nodes move.

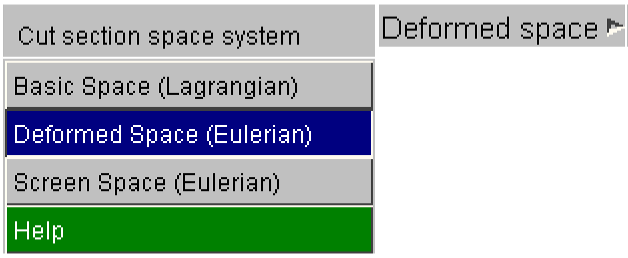
9.5.4. Defining a Space System for the Plane

Defining a space system for the plane.

Once you have defined the plane, by one of the definition methods above, you need to define which space system it operates in.

This figure shows the Cut space system selection panel, showing the three possible systems. These are described below.

Section follows node(s) allows a cut section defined using 3 nodes, or a single node in the constant X/Y/Z cases, to be updated using the current coordinates of the node(s) at each state.



BASIC space system

In this system the cut plane is calculated using the model's **undeformed** geometry, regardless of the current state in core.

This means that the parametric coordinates of the cut positions on elements are calculated using the undeformed geometry, then applied to the current (deformed) in-core state. Therefore the cut plane will almost certainly not remain flat as the model deforms.

This is a "lagrangian" cut: the cutting plane deforms as the element mesh deforms.

DEFORMED space system

In this system the cut plane is calculated using the model's current **deformed** geometry. Therefore the cut position on elements, and indeed the elements which are cut, can change as the model deforms through the static plane position.

The plane will always remain flat, and will remain fixed in space relative to the model coordinate system.

This is an "eulerian" cut: the cutting plane remains fixed while the element mesh can deform through it.

SCREEN space system

In this system the cut plane is calculated using the current screen coordinates, after the transformation and projection to screen space.

This has the effect of tying the cutting plane to the screen space system, effectively to your display, therefore both deformations and viewing transformations (e.g. dynamic viewing) can move the model through the plane.

This is also an "eulerian" transformation since the model deforms through a static cutting plane.

Note: Force and moment computation varies with section space.

For compatibility with LS-DYNA the forces and moments computed in a BASIC space system are:

- Always expressed in the global cartesian system
- Centred on the average coordinate of the cut section at each state.

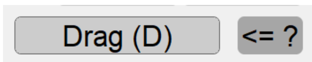
Whereas those computed in a DEFORMED or SCREEN space system are:

- Always expressed in the section local coordinate system.
- Centred on the plane origin as defined by the user.

This is described in more detail in FORCES Computing forces and moments on the cutting plane below.

9.5.5. Dragging a Plane Interactively Using the Mouse

Dragging a plane interactively using the mouse



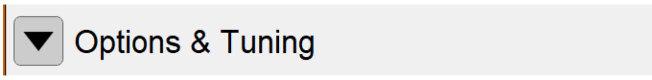
The " **Drag** " button (or the "D" keyboard shortcut) switches D3PLOT into cut-section plane dragging mode.
In this mode the cursor symbol changes to "Sect Drag", and the mouse buttons act as follows:

- Left mouse ***translates*** the plane in the normal (local **Z**) direction.
- Middle mouse ***rotates*** the plane about the plane local **X** axis
- Right mouse ***rotates*** the plane about the plane local **Y** axis.

These operations are chosen because they are the most commonly used "drag" functions.

When planes are defined in multiple directions, the dragging changes that plane whose direction tab is currently selected on the panel.

The **Options & Tuning** button gives a more comprehensive plane dragging sub-menu.



You choose either Translate or Rotate, and in each case

- Left mouse translate along / rotates about the **X** axis
- Middle mouse ... ditto ... about the **Y** axis
- Right mouse ... ditto ... about the **Z** axis.

You can choose whether these translations / rotations use plane local or model global axis systems.

Cut section summary forces are shown as in the master panel and, if selected, will be updated as the plane moves.

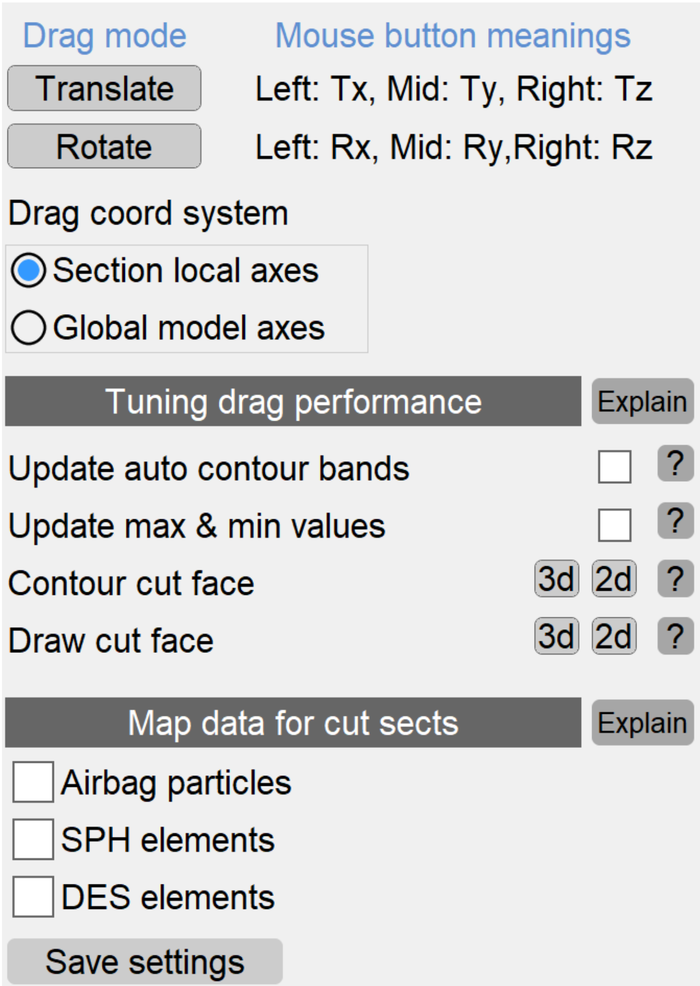
Tuning Drag Performance

Dragging a cut section through a big model can be slow, especially if the current plotting mode shows contours and the current levels are in "automatic" mode.

The options in this panel allows you to alter the behaviour of cut sections when they are dragged with the mouse giving a trade-off between image appearance, contour accuracy and speed. The [?] buttons against each option give details of each setting.

These settings only apply during the "drag" process itself, once you release the mouse button to end the drag the image will be redrawn showing the missing graphical information.

The current settings can be saved as preferences in the oa_pref file by using the "Save settings" option.



9.5.6. Section Follows Origin Node(s)

Section follows origin node(s)

If a cut section has been defined using either the 3 node method or a single node and a global axis system then this option will force the cut section to follow the node(s) as they move during the analysis.

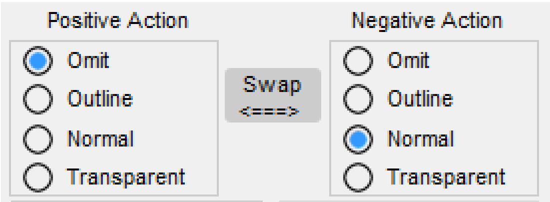
Where there is more than one model in the window(s) affected then the following rules apply:

- For each model the labels of the nodes are looked up.
- If all the necessary nodes (3 in "3 nodes" case, 1 in "constant X/Y/Z" case) are found then the normal logic will apply based on the current coordinates of the nodes in each model, and the section will be updated at every state.
- If a node is not found then the "follow" logic is turned off for that model, and the plane will remain static in its initial position for that model.

Note that using this logic over multiple models may mean that the planes in each model may not be the same, as the defining nodes may move differently. Exercise care using this option!

9.5.7. Positive and Negative Action (Controlling Display of Structure Either Side of the Cutting Plane)

Positive and Negative Action (controlling display of structure either side of the cutting plane)

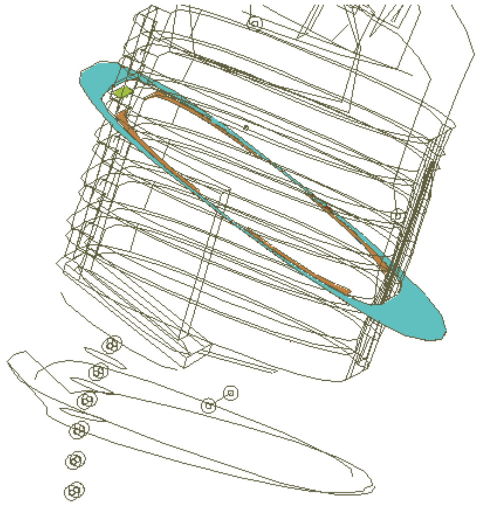


By default when cutting planes are switched on only the cut elements are drawn normally, with the remainder in wire-frame.. But it is possible to draw the mesh on both positive and negative sides of the plane at three levels of complexity. Each side can be controlled separately.

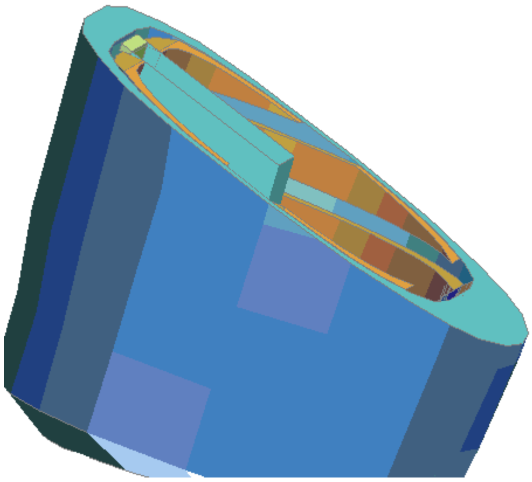
The options are:

OMIT	The mesh on this side is not drawn at all.
OUTLINE	The mesh is drawn in "line" mode. This means no hidden-surface removal, and the cut plane will be visible through the mesh.
NORMAL	The mesh on that side is drawn normally, with contoured data if applicable. Contours will be continuous over cut and uncut faces.
TRANSPARENT	The mesh on that side is drawn in a transparent mode, with contoured data if applicable. Contours will be continous over cut and uncut faces.

This image shows a cut section with the default settings(with both sides in outline mode)



This image shows a cut section set to Omit on the positive side of the cut and Normal on the negative.

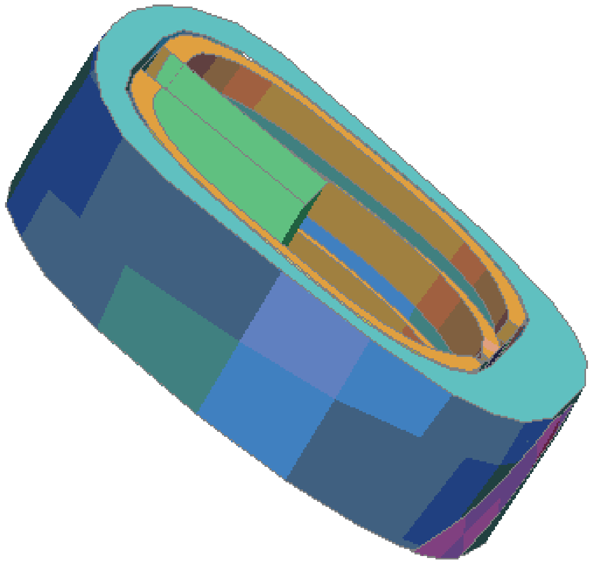


9.5.8. THICK CUT Creating Cut Sections with a Finite Thickness

THICK CUT Creating cut sections with a finite thickness

Normal cut sections represent an infinitely thin slice through a model. The **THICKNESS** option can be used to generate a finite thickness cut through a model.

A 75mm thick cut



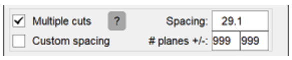
9.5.9. MULTIPLE CUTS Creating Multiple Parallel Cut Sections

MULTIPLE CUTS Creating multiple parallel cut sections

Normally only a single cut section is created, however you can choose to create multiple parallel sections by enabling the **Multiple Cuts** tick box.

9.5.9.1. Uniform Spacing

Uniform spacing

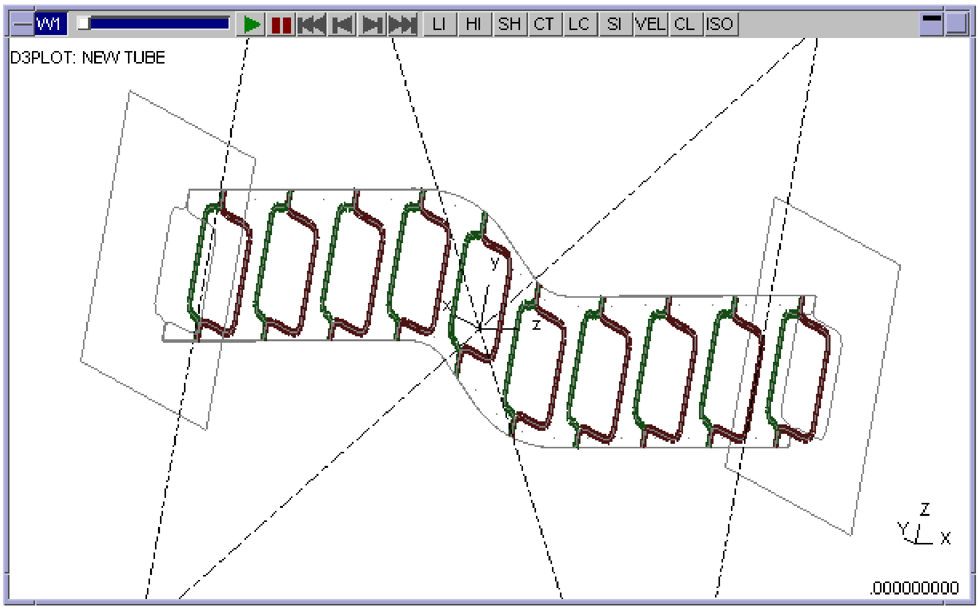


When **Custom spacing** is off, these multiple parallel cut planes are located at a constant spacing either side of this "base" section.

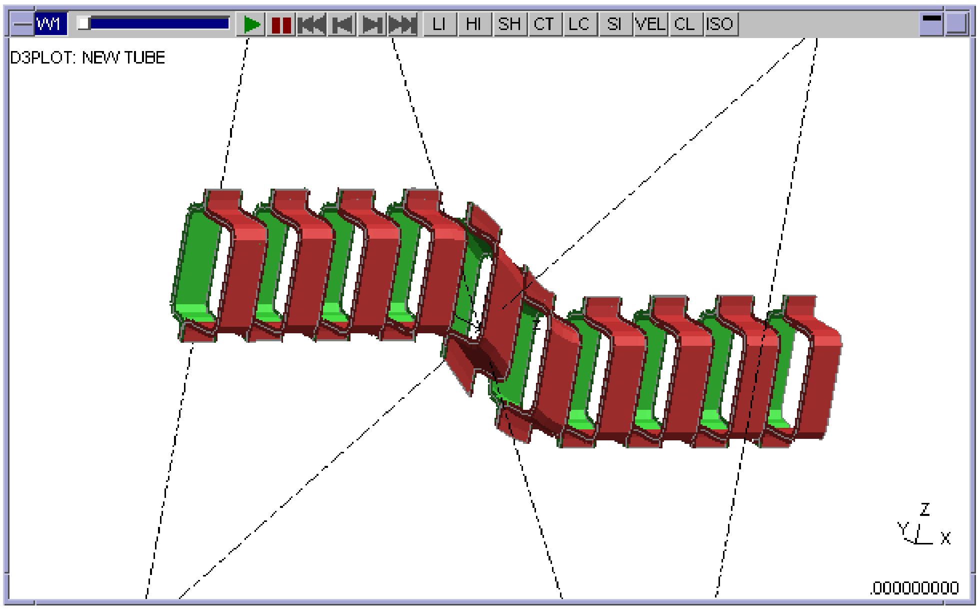
Choose the **spacing** between planes. The default value is approximately 10% of the largest diagonal of the bounding box around the model.

By default cuts will extend the full distance on either side of the base plane to include the whole model, subject of a "sanity check" limit of 999 planes on each side. You can limit this by setting the numbers **# planes +/-** on +ve and -ve sides, both values being in the range 0 - 999.

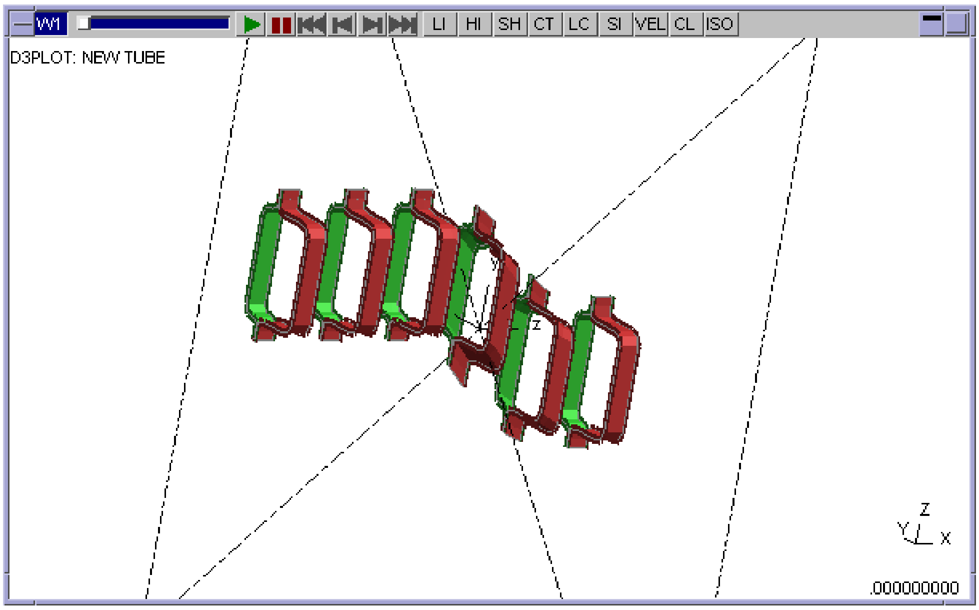
Here is an example of a multiple plane allowed to repeat the default number of times, so that it spans the whole model from end to end. Wireframe plane display has been switched on to show the "base" plane.



Here "thick cuts" have been turned on for the example above, showing how they can be used in conjunction with multiple planes.

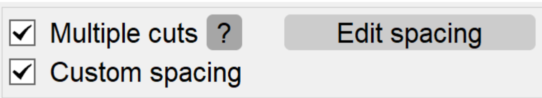


Here the number of planes has been limited to 3 on the -ve side and 2 on the +ve side,showing 6 in all (as the base plane is always drawn). The thickness of the sections has also been reduced.

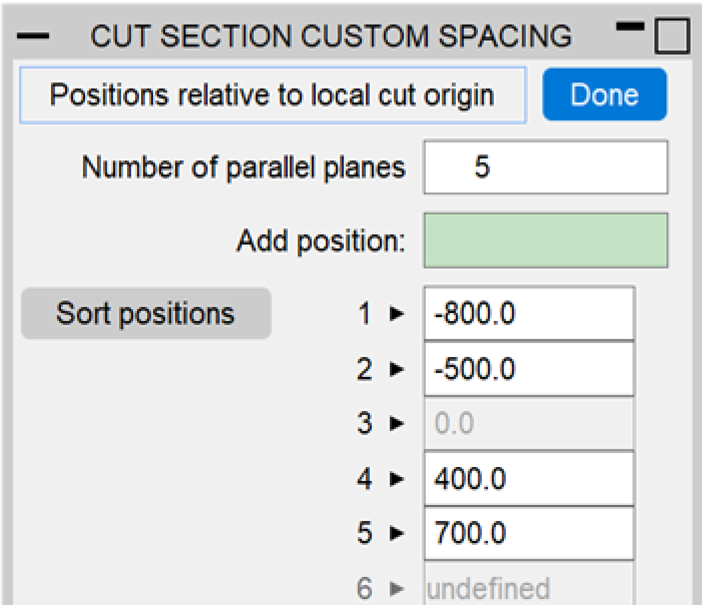


9.5.9.2. Custom Spacing

Custom spacing

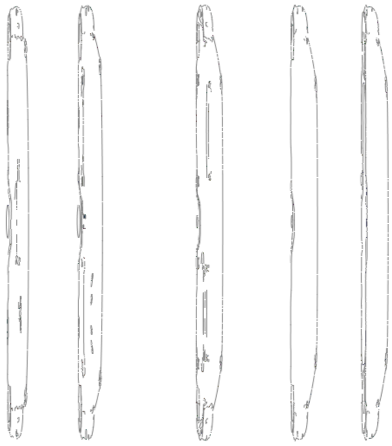


When **Custom spacing** is on, any not necessarily uniform sequence of offsets from the base plane can be defined. By default there are two additional planes shown on each side of the base plane, so five planes in total, but with the **Edit spacing** button each plane position can be edited individually. The **Sort positions** button can be used to sort the plane positions into ascending order.



We can also turn on thick cuts while using custom spacing.

See the comparison below when "Thick cuts" (with thickness is set to 60.0) is OFF and ON respectively:



9.5.9.3. General Remarks

General remarks

Parallel cut planes can be used with both Basic and Deformed space, and may have contours displayed on them. However you should note the following:

- Generating the graphics for multiple planes can become quite slow if many cuts are made through a large model. This simply because it requires a lot of maths to calculate all those slices, so don't be surprised if processing these sections is slow.

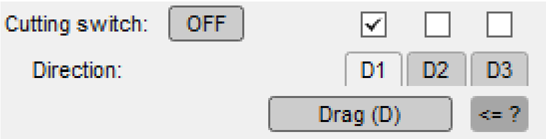
- Once computed the rendering of multiple "thin" planes should be reasonably fast, however multiple "thick" sections may be quite slow to render since the hardware is having to do a lot of clipping calculations each time the display is updated.

- Screen-picking from multiple sections is also difficult since, in theory, an element could be cut many times giving many potential candidate locations for selection. Therefore screen-picking is only approximate when multiple sections are in use, and while it should find "cut" elements it may occasionally also select elements that are not visually correct. If this happens try moving the selection point a little.

9.5.10. Multiple Non-Parallel Cuts

Multiple directions for non-parallel cuts

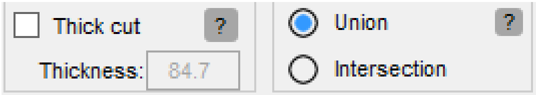
Cut planes can be defined in up to three directions. By default only one plane direction is active. However, it is possible to have any combination selected (this includes none or all three directions). Enabling the cut plane for each direction can be managed with the tick boxes above each direction tab **D1**, **D2**, and **D3**:



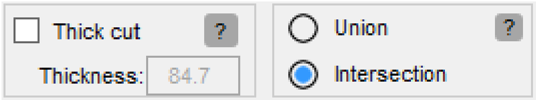
Note that the cutting switch applies to all plane directions together. When it is switched off, the graphics window(s) show the model(s) as usual. When it is on, then the intersection or union of cut planes (depending on the enabled mode) with the model with all planes defined in these tabs is shown. The properties for each cut plane direction can be modified by selecting the relevant direction tab.

Union and Intersection Mode

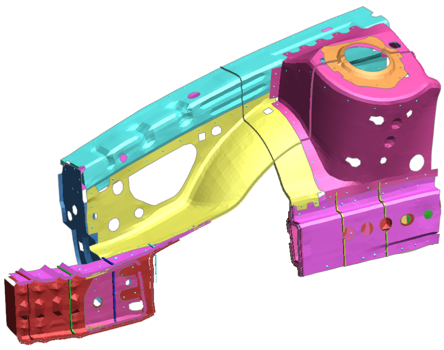
There are two modes when combining multiple parallel cut sections: **Union** and **Intersection** modes. By default, Union mode is selected. However, changing between the two is possible using the radio buttons located on the cut section panel. See the image below for an example.

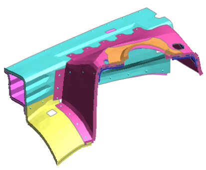


When intersection mode is selected, the enabled cut planes will only show the elements that intersect. See the image below for example.



The two images below are a result of 2 intersecting orthogonal planes with Omit and Normal being used on both directions. Union mode (left) and Intersection (right) modes are both shown for comparison.

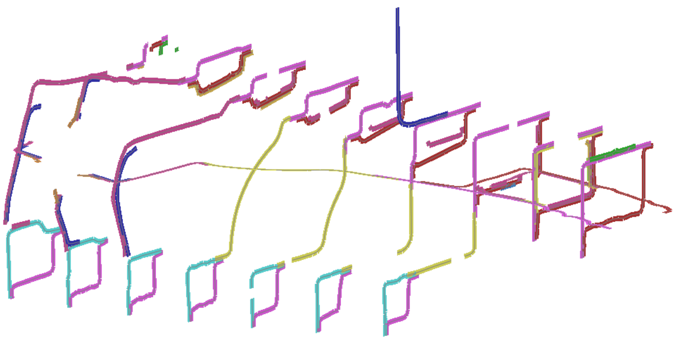


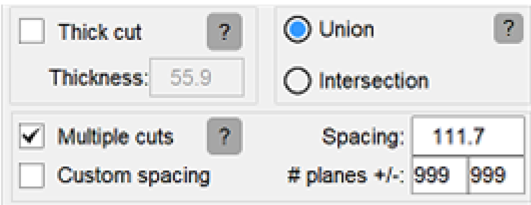


Note that as of D3PLOT 20.0, Intersection mode is not available for cuts in Basic Space.

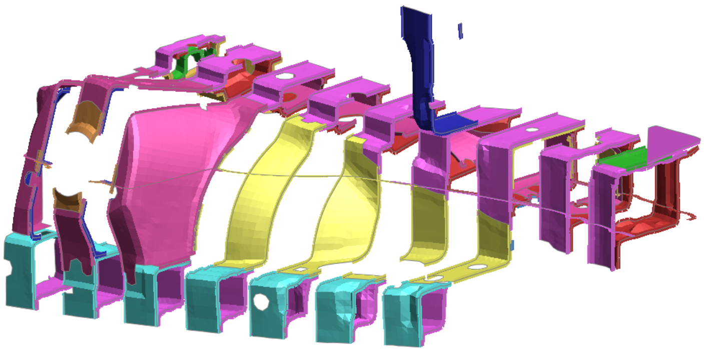
Combination with Other Settings

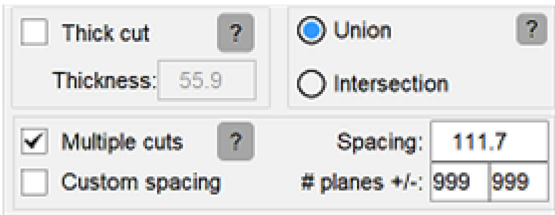
It is possible to combine the multiple non-parallel cut sections with other options for each direction. As an example, multiple parallel cuts can be activated for each plane direction individually. Either uniform spacing or custom spacing is available. The following is an example with uniform spacing multiple cuts for the first direction and no multiple cuts for the second direction. Omit and normal are used for the positive and negative actions:





Additionally, thick cuts can be enabled when multiple plane directions are active. Below is an image of a cut section with two orthogonal planes with union mode enabled. One cut direction has thick cuts enabled.

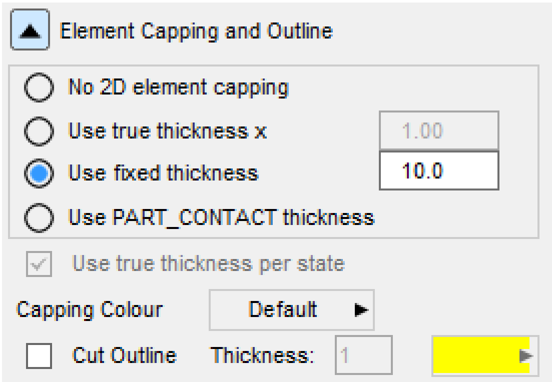




9.5.11. Element Capping and Outline

Element Capping and Outline

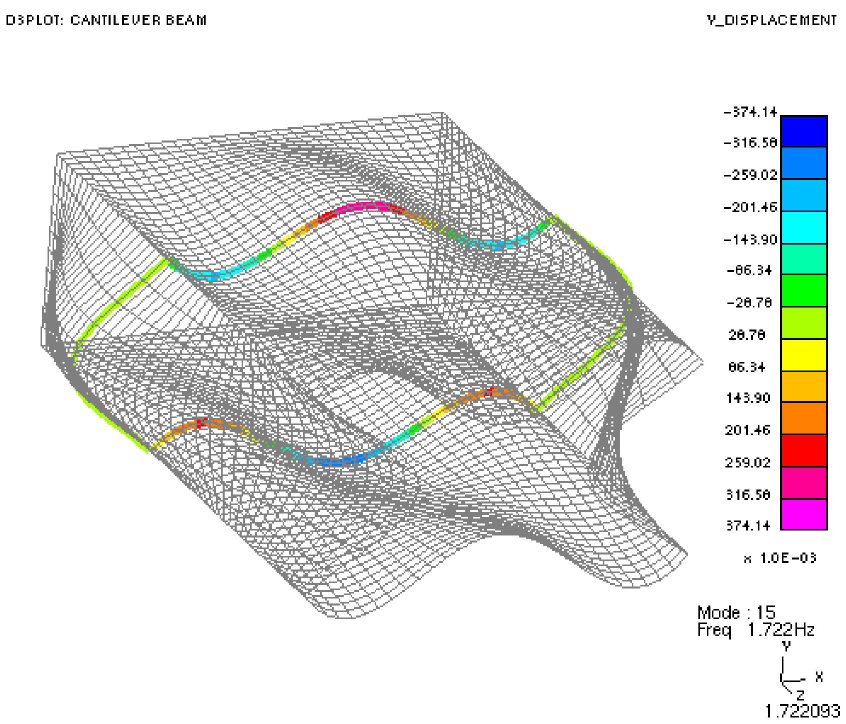
The **Element Capping and Outline** portion of the Cut Section menu controls 2D element capping properties as well as cut section outlining. Various controls for thickness and colour are available for both, as shown in the figure below. Default settings are shown here:



Element Capping

Thickness

There are controls to determine whether or not 2D elements (shells and contact segments) have "caps" drawn where they are cut. A "cap" is a thick line where each 2D element is cut, and it will show the current visual properties of the element, as in the example below:



This example shows an eigenvalue analysis of a hollow box of shells. A cut section of constant Z has been applied half way along, and contours of Y displacement drawn. The "capping" of the shells shows the displacement around the cut section.

The default thickness of 2D element caps is a fixed thickness, which can be adjusted. In this case a value of 10.0 gives a satisfactory plot.

In some cases, you may wish to plot the true thickness of the shells and this option can be selected. The displayed thickness of the shells may be factored by changing the value of 1.00 to a larger or smaller value. By default, the true thickness is taken as the shell thickness at state 1 and is not updated for other states. If there is a requirement to updated the true thickness at each state then this option can be selected.

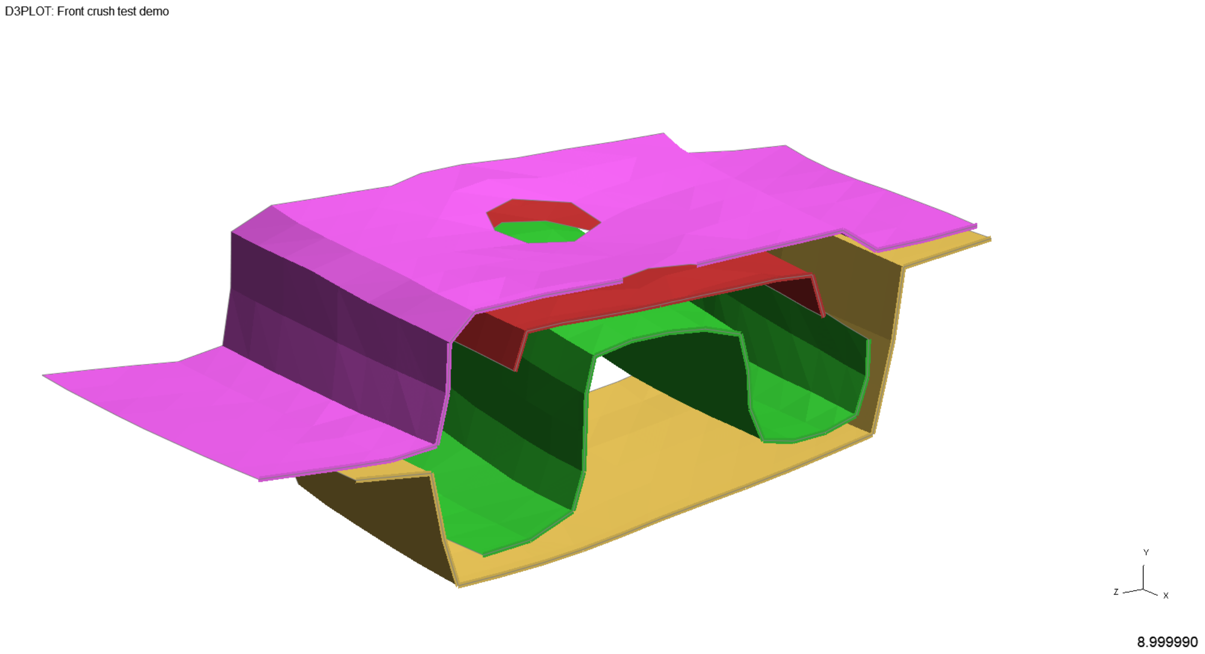
If **Use PART_CONTACT thickness** is selected used then shells with a *PART_CONTACT card will use the following as their capping thickness:

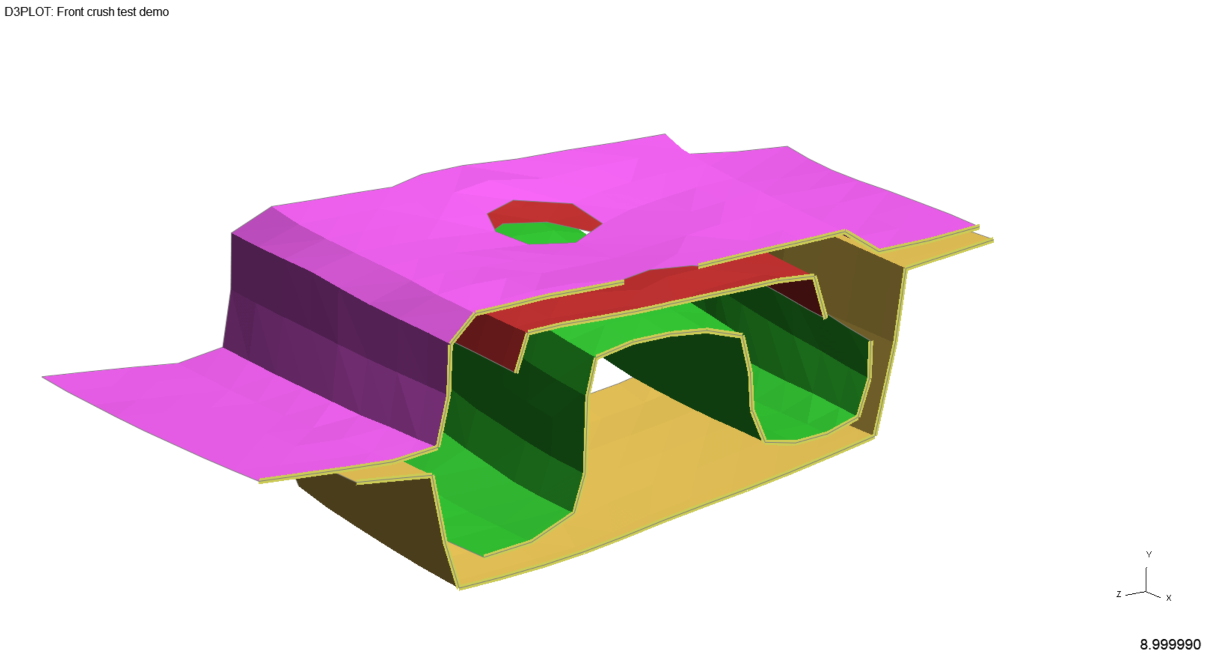
- If OPTT is non-zero that is used verbatim
  else
- If SFT is non-zero then (SFT * true shell thickness) is used

If neither OPTT nor SFT are defined, or the shell's part is not _CONTACT, the unfactored true thickness is used.

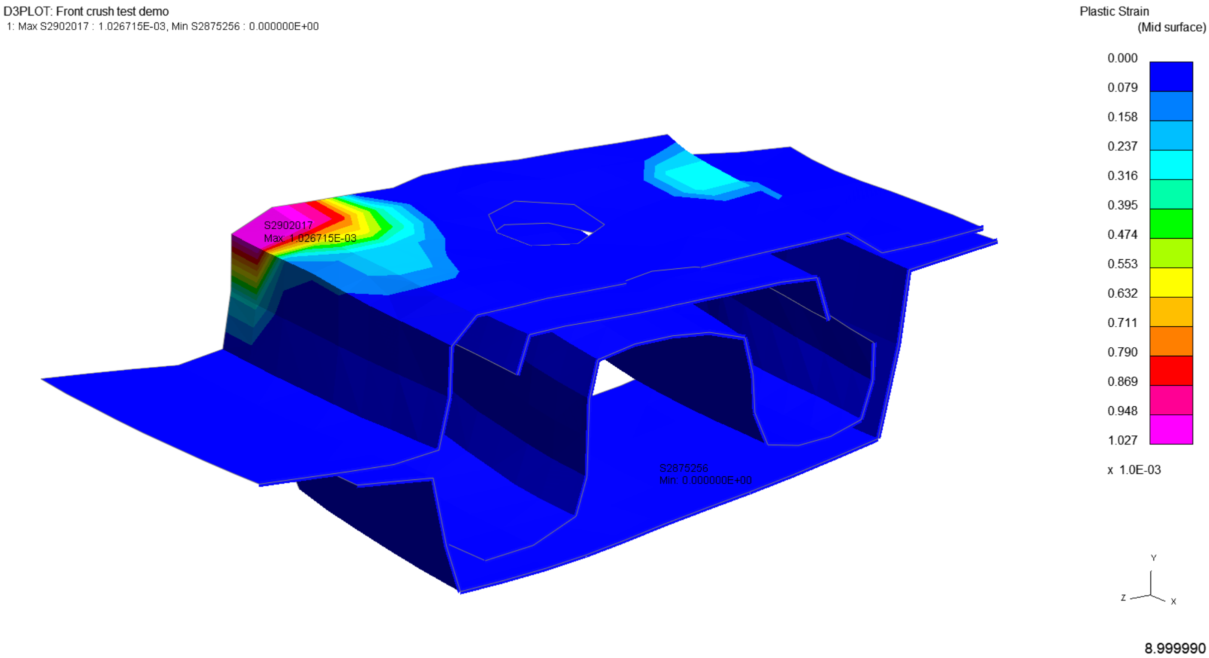
Colour

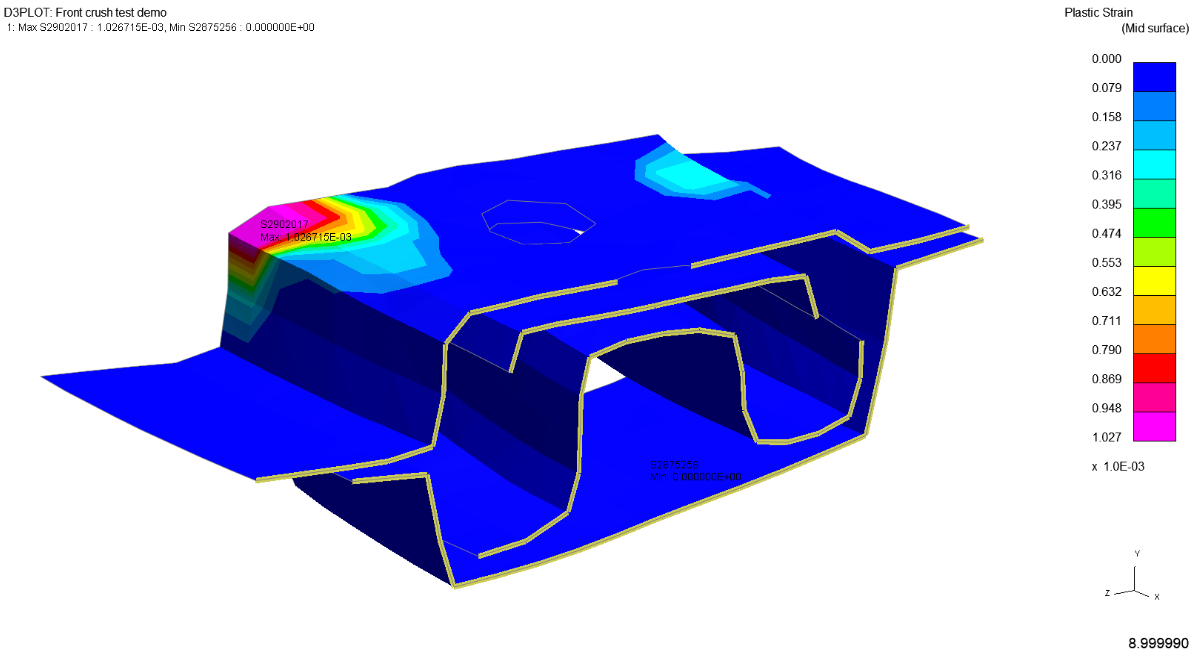
This next example demonstrates the usage of the capping colour controls. This allows for easier visualisation of the cut edges. There is a choice to choose between a core/custom colour or "Default". By choosing "Default", the default entity colour will be used in Shaded, SI and CT. Below is an example comparing the difference between "Default" (left) and selecting the colour yellow (right) for shaded plots.





Below is an example comparing the difference between "Default" (left) and selecting the colour yellow (right) for a SI plot.

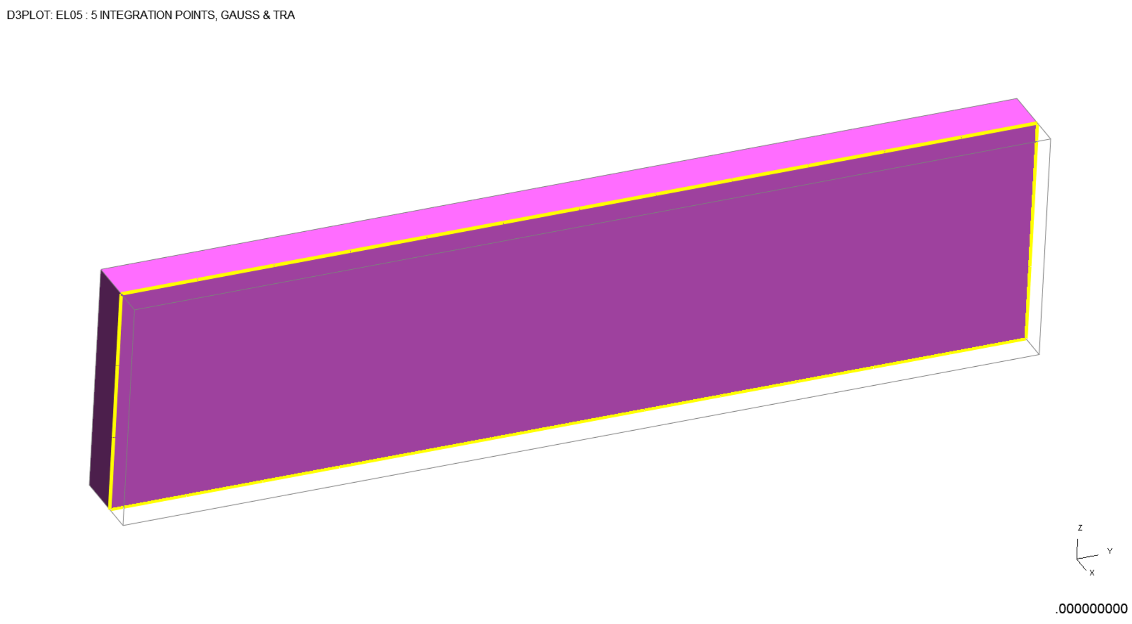




Cut Outline

Thickness and Colour

The cut outline can be controlled to make it easier to see the where the cut is applied. This is an alternative to controlling the colour of the element capping, since element capping is only available for 2D elements. The cut outline is currently available for shells, thick shells and solids. It can be toggled on and off and a thickness integer value between 1 (pixel) and 10 (pixels) can be applied. Core and custom colours can be applied. Shown below is an example of a cut solid element with a custom yellow outline and thickness value of 5 pixels.



9.5.12. Exclude

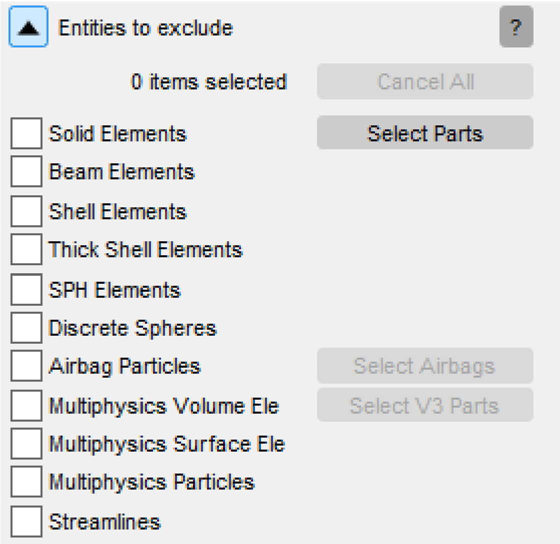
Exclude

By default D3PLOT will apply the cutting plane to all the Solid, Beam, Shell, Thick Shell, SPH, DES, Airbag particle and Multiphysics solve elements.

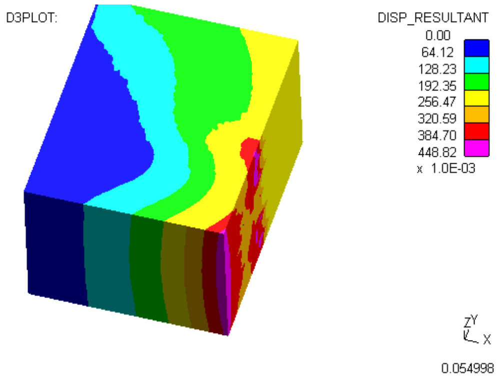
This option can be used to exclude individual element types from the cutting plane so they can be seen on either side of the plane regardless of the +ve and -ve action settings.

In addition, individual Parts, Airbags and Multiphysics (Volume III) Parts can be selected to be excluded.

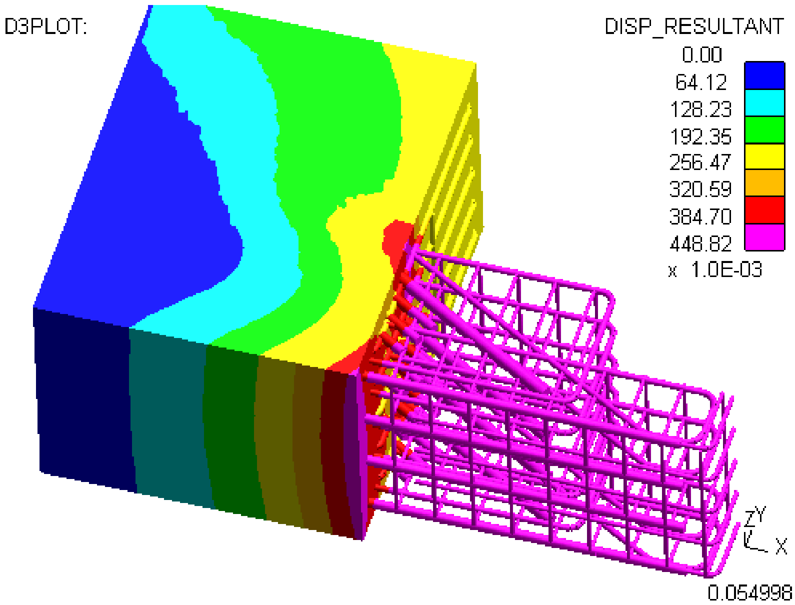
If this option is used then it should be noted that the calculation of cut section forces ignores this option and will include any unblanked elements that are cut by the plane.



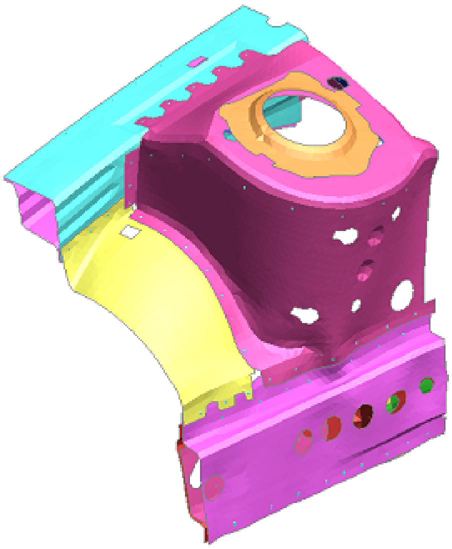
The example opposite shows a block of solid elements (representing concrete) that has been cut in half using a cut section.



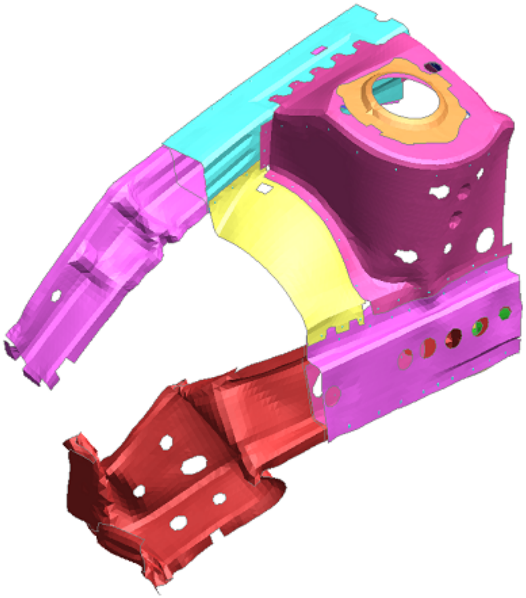
By turning on the option to ignore beam elements the beams that represent the steel reinforcement bars within the concrete become visible.



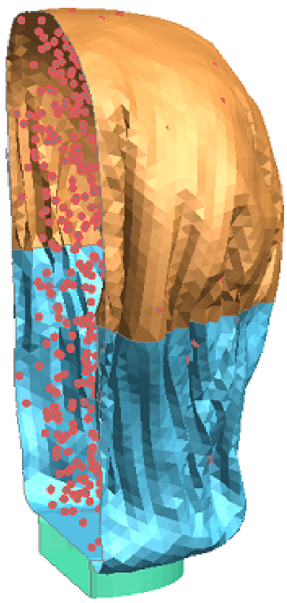
Here is another example showing some structure cut by a section.



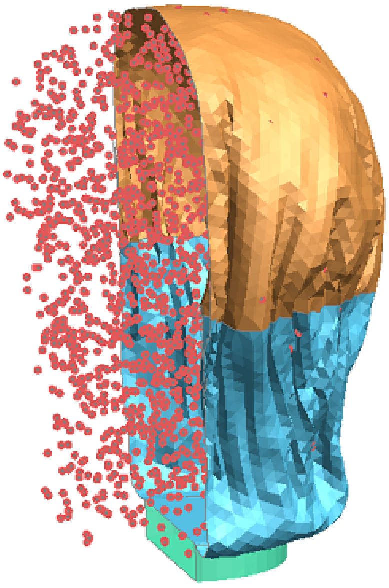
By selecting some parts to be excluded from the cut section they are made visible.



Here is another example showing an airbag cut by a section.



By selecting the airbag to be excluded from the cut section the airbag particles are made visible.



9.5.13. FORCES Computing Forces and Moments on the Cutting Plane

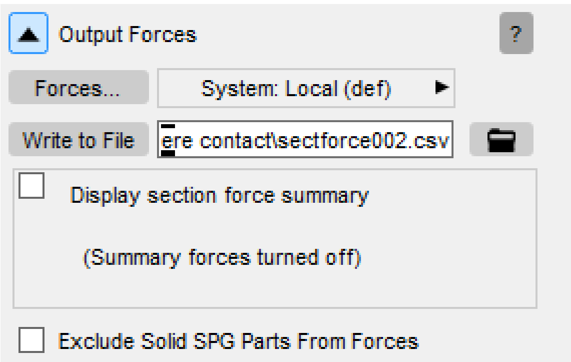
FORCES: Computing forces and moments on the cutting plane

The **Output Forces** portion of the Cut Section menu calculates the forces and moments acting on the current current plane.

The **Write to File** button will write the forces and moments to a CSV file.

The **Forces...** button brings up a comprehensive table of cut section forces data, with options to write to Text, CSV, or XLSX file format.

A summary of the cut section centre coordinates and the current forces for the current direction tab can be shown in the panel if the check box there is ticked. The values will update automatically as the section is dragged.



Please read the following section on force and moment extraction before using this facility. There are some less than obvious pitfalls that you need to consider. In particular:

- Only ***unblanked*** elements are included in cut section force calculation.
- Any element types or parts excluded from the cut section are not included in the force and moment calculation.
- Rigid elements may be transmitting force, but they will always report zero output to the database. Therefore their contribution to cut forces and moments will always be zero.
- Forces and moments are only computed from Solids, Beams, Shells and Thick shells. Other element types either do not report forces (eg springs, seatbelts), or are not sensible in this context (eg SPH elements).
- There are inconsistencies in the way LS-DYNA writes beam force and moment output prior to LS971, requiring user intervention if the correct answers are to be calculated.
- Local bending moments in thick shells are not included, and may also be omitted for thin shells if force & moment resultant data components are not present in the database.
- LS-DYNA processes cut-sections in "basic" space, generating forces and moments in the global system. If you switch to "basic" space in D3PLOT you will get a similar calculation, but "deformed" space results in D3PLOT are expressed in the plane's local system.

9.5.13.1. How Cut Forces are Calculated

How cut forces are calculated

Only forces in the following ***unblanked*** element types are computed: solids, (thin) shells, thick shells and beams. Other element types (e.g. springs) are ignored since LS-DYNA does not report forces in them in a way that can be read by D3PLOT. The force and moment values are integrated from the element stress & force results as follows:

Solids:

The cut face through the solid is interpolated, and its area calculated. The element stress tensor is rotated to the cut plane system and the forces are calculated from:

$F_X = \tau_{XZ} * Area$	Where:	Fx is in-plane X force
$F_Y = \tau_{YZ} * Area$		Fy is in-plane Y force
$F_Z = \sigma_Z * Area$		Fz is normal Z force

No local element moments are calculated within solids: they are constant stress elements.

Fully integrated solids with more than one integration point.

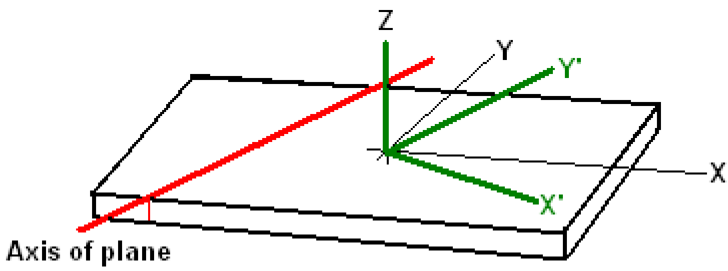
Fully integrated solids are, of course, not constant stress elements and they can support bending moments. However by default these element types only report averaged results for a single integration point at the element centre to the PTF file meaning that they are still effectively constant stress elements with no bending for the purposes of post-processing.

It is possible to write data from all 8 points to the PTF file, and D3PLOT will read these results, however support within the code for this is very limited and does not currently extend to calculating local bending. This issue will be dealt with in future releases.

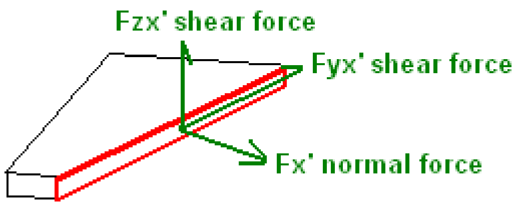
Thin shells:

The forces are calculated using the shell force resultants [`Fx,Fy,Fxy,Qzx,Qzy`] which yield a stress tensor in the shell local coordinate system when divided by shell thickness.

The element local stress tensor is rotated about element local Z axis to align it with the cut axis (red), giving a new system [X', Y, Z'] (green) where Z' is the same as element local Z.



The local stresses can, when multiplied by the cut area (red) give forces acting on the plane.



`Sx'`	* cut area	gives normal force on plane	`Fx'`
`Tyx'`	* cut area	gives transverse shear force	`Fyx'`
`Tzx'`	* cut area	gives vertical shear force	`Fzx'`

The cut plane is always treated as cutting the element "cleanly" in the element local Z direction, this is true even if the axis of the plane is sloping and the cut is oblique. Therefore the cut area is always (cut length * shell thickness) regardless of the obliquity of the cut..

This force vector [`Fx'`, `Fyx'`, `Fzx']` can then be rotated to the cutting plane system, taking into account signs.

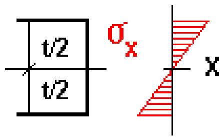
Local element moments are also obtained by rotating the moment resultants `[Mx,My,Mxy]` to the cut plane axes.

Warning : you should take care to distinguish between the Timoshenko convention local moments derived from stresses (`Mx`, `My`, `Mxy`) as described below, and the bending moments acting about the cut plane axes (`Mxx`, `Myy`, `Mzz`).

`Mx` = local bending moment per unit width due to local `X` direction stress.

This gives rise to bending term ***`Myy`*** about the element local Y axis.

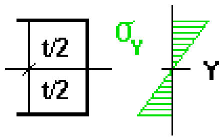
$$M_X = \int_{-t/2}^{+t/2} t.\sigma'_x.dt$$



My = local bending moment per unit width due to local `Y` direction stress

This gives rise to bending term ***`Mxx`*** about the element local X axis

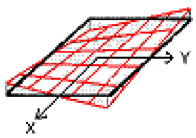
$$M_Y = \int_{-t/2}^{+t/2} t.\sigma'_Y.dt$$



`Mxy` = local torsion (warping) moment per unit width due to local `XY` shear stress

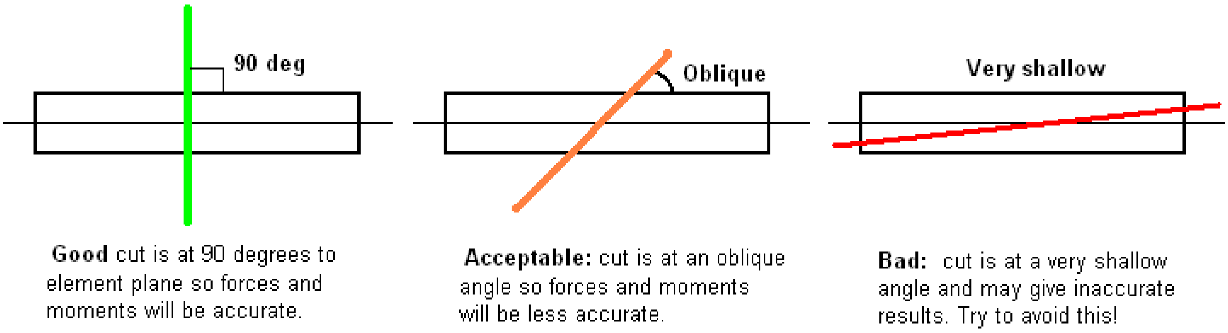
This gives rise to torsion terms `-Mxx` and `+Myy` about the element local X and Y axes

$$M_{XY} = \int_{-t/2}^{+t/2} t.\tau'_{XY}.dt$$



The signs of the local bending moments ***`Mxx`***, ***`Myy`*** and ***`Mzz`*** take into account the orientation of the cut plane with respect to the element local axes.

Shells calculate stresses in the (thin) plane of the shell, and do not develop a full 3D stress state. Therefore forces and moments in shells are reasonably accurate so long as the cut plane intersects the element "cleanly" at something close to 90 degrees as shown the the diagram below. Oblique cuts will still give reasonable results so long at the cutting angle is not too shallow. Very shallow cuts may "fall off" the edges of the element as shown on the right below, and give misleading results.



If shell force and moment resultants are not present in the database then the element neutral axis stress tensor is used instead. However

- ***Only the mid-surface results are used*** , treating the element as plane stress, which means that ***element local bending moments are not calculated*** . (*Since the location of the element integration points and the degree of plasticity are both unknown it is impossible to calculate an accurate local bending moment.*)

Thick shells:

Forces for thick shells are calculated by rotating stresses within the element about the element local Z axis to align with the cut plane. The cut plane is assumed to be cutting at right angles to the surface, so for best results, the cut should be close to at right angles to the surface. Local moments are obtained by rotating the moment resultants to the cut plane. Each element contributes its own individual bending moment due to the bending stresses in the element as well as the bending moment on the cut plane due to its section forces multiplied by the distance to the cut plane origin. The way forces and moments are calculated is essentially the same way as in thin shells with some differences detailed below (for more details, read the thin shell section just above this).

In thick shells, forces are calculated using the stresses at each integration point whereas for thin shells force values are extracted directly from the LS-DYNA binary output. Thicknesses are calculated using the node positions instead of being taken from the element properties. It also should be noted that force values displayed in plots are per unit length, so the cut length across each element is taken into consideration. If the element is not uniform, the length used to estimate forces is the average of the cut across the top and bottom of that element.

It should be noted that for 6 noded thick shell elements, cut section force and moment values are less accurate (compared to 8 noded thick shell elements). Also, if forces at each integration point cannot be extracted (a likely cause is a missing ztf file), the cut force will be estimated using the mid surface stress and moments calculated will be 0. The reason moments will be shown as 0 is because we would have to assume a linear stress distribution (implicitly linear elastic) which is likely to be wrong. We believe that it is better to compute no moments than a plausible but wrong one.

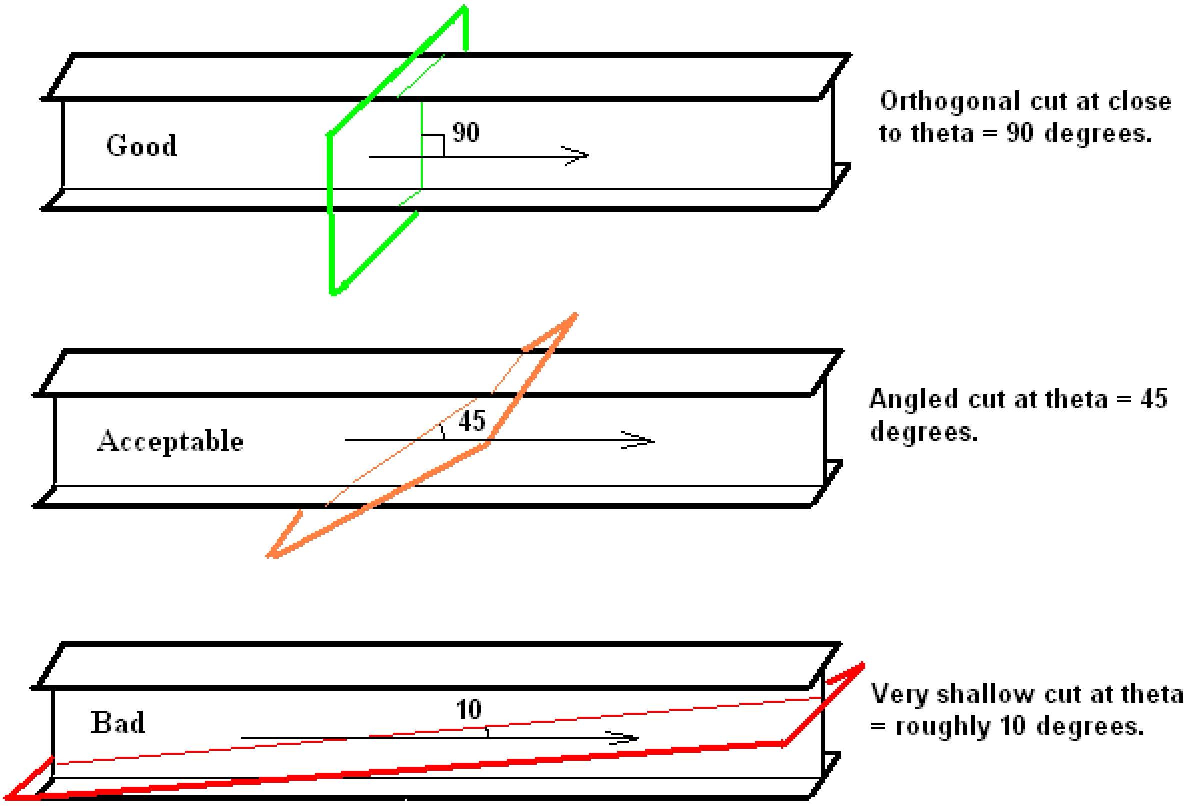
Beams:

Forces :

The three beam forces normally written [`Fx,Fy,Fz`] are actually a direct axial force `Fx` , and two transverse shear forces which should really be written `Fyx` and `Fzx` : (as in shear forces in Y and Z respectively on a plane of constant X).

This local force vector [`Fx. Fy, Fz`] is rotated into the axis system of the cut plane, taking into account signs, to give a normal force and two shear forces in the cut plane system.

Forces and moments in beams are reasonably accurate so long as the cut plane intersects the beam cleanly at roughly 90 degrees.



.Moments : These are straightforward:

- The moments [**Mxx,Myy,Mzz**] are treated as a vector, and are rotated from beam local to cutting plane system and used directly.

Reporting the cut plane centroid in models containing beams

D3PLOT is able to calculate the total area cut through all elements, and hence the cut centroid (average coordinate), which enables it to report cut section results in the same "basic space" coordinate system as LS-DYNA.

For solids and shells this calculation is performed by calculating the first moment of the cut area and then dividing through by this area to obtain a centroid, but this calculation cannot include beam elements since their cut-section area is usually not known.

Therefore the following procedure is adopted when a cut plane intersects beams:

- If both beam and other (solid and/or shell) elements are cut then the cut area and centroid is based on the cut area through the solids and shells only.
- If only beams are cut then each is assigned a notional area of 1.0, and the cut centroid will be the average coordinate of all the cut beams. In this situation a

cut area of 1.0 is always reported, regardless of the number of beams, in order to make it clear that the value is not "real".

Inconsistent beam sign conventions in LS-DYNA releases up to and including 970

Due to a bug in LS-DYNA versions up to and including LS970 exhibit the following inconsistent sign convention for beam output:

- "Resultant" (typically Belytschko-Schwer) elements use one sign convention
- "Integrated" (typically Hughes-Liu) elements use the opposite sign convention for 4 of the 6 output components.

The following table shows the sign conventions from releases 970 and earlier:

Component	Matching?
Fx	Same
Fy	Opposite
Fz	Opposite
Mxx	Opposite
Myy	Opposite
Mzz	Same

Sadly there is no "right" convention for beam output, as different users have different conventions. The confusion arises because of the different ways in which the beam types work: integrated beams have integration points at their centre, whereas resultant beams have (potential) hinges at their ends. The former reports force in the beam, and the latter reactions at the supports.

D3PLOT attempts to draw bending moment diagrams on the tensile side, but depending on which beam type you have used this may or may not be the case.

Beam sign conventions are consistent from LS-DYNA release 971 onwards

At some stage during the development of LS971 this problem was fixed, and results now use the "integrated" convention for all beam types. This is consistent with the reporting method for other element types in LS-DYNA, where results are the forces and moments within the element.

How D3PLOT handles the beam sign convention problem

The sign convention is crucial when computing cut forces, since the force and moment vectors are transformed into the plane of the cut, and a reversal of their sign obviously affects the answers.

Unfortunately D3PLOT can't tell from earlier results files whether an output database is from LS-DYNA 971 or later, since although the database contains a "version" field LS971 writes "970" in there! Therefore it doesn't "know" which sort of beam it is dealing with and it will ask you what beam types you have used when you first calculate cut forces through a structure. Thereafter it will apply correction factors as required . If you have mixed the two beam types in your model you will have to be extremely careful when interpreting results from a pre-970 analysis.

If you are not asked to define a system then your results file is from a version of LS-DYNA 971 onwards that is recent enough to encode up to date version information, and D3PLOT has been able to determine its format automatically.

WARNING:	**Rigid elements report zero stresses, although they may still be transmitting loads.** The cut forces in these elements will be calculated as zero.

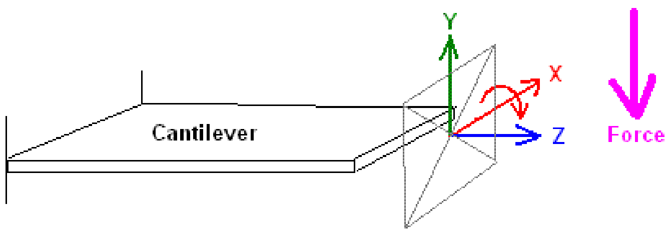
9.5.13.2. Sign Convention of Cut Forces

Sign convention of cut forces

The sign convention adopted is that of the **forces acting on the plane from its positive side** , expressed in:

- **Deformed space** : the plane's local coordinate system.
- **Basic space** : the global model system

Moments are expressed about an axis, using the right hand screw rule. Thus Mxx is moment about the plane X-X axis in the relevant system, with a +ve moment being generated by clockwise twist looking from the origin down that axis.



In this example the cutting plane is shown in grey, with its axes triad in red (X), green (Y) and blue (Z). A force applied "downwards", here in the -ve plane Y direction, on the +ve (Z) side of the plane generates

a +ve moment about plane X.

a -ve shear force in plane Y

This figure illustrates how the sign convention affects moments. Since the force above is "acting on the plane" from its outside it generates a +ve moment about the plane XX axis.

If the plane were rotated 180 degrees about its local X axis, reversing "behind" and "in front" sides, then the sign of the moment about XX would be negative. In addition the plane Y axis would now point downwards, and the shear force in Y would be +ve.

9.5.13.3. Coordinate System and Centroid of Cut Forces

Coordinate system and centroid of cut forces

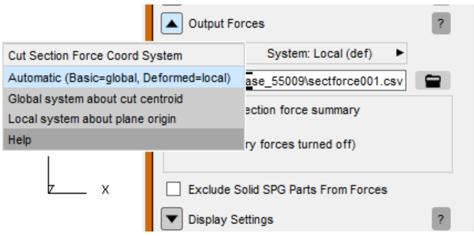
The coordinate system and centroid depend upon the plane's space system:

<table>
<tr><td colspan="3">In DEFORMED space the following is used, all axes being in the plane local axis system .</td></tr>
<tr><td>Fx is in plane X force (effectively shear force)</td><td>Mxx is moment about plane local XX axis</td><td rowspan="3">The plane centroid is at the plane's origin, and its local X, Y and Z axes are as defined by the user. These axes may be visualised by turning plane display on. Unless "cut follows nodes" is turned on the centroid and axes remain fixed as the model deforms.

In particular note that moments are calculated about plane local axes acting through the plane origin.</td></tr>
<tr><td>Fy in is plane Y force (also shear force)</td><td>Myy is moment about local YY</td></tr>
<tr><td>Fz is Z force normal to the plane.</td><td>Mzz is moment about local ZZ</td></tr>
<tr><td colspan="3">In BASIC space the following are used, all axes being in the global model system .</td></tr>
<tr><td>Fx is force in the global X axis</td><td>Mxx is moment about the global XX axis</td><td rowspan="3">The plane centroid at any given state is the average coordinate of the cut elements, this means that it moves as the model deforms.

In particular note that moments are calculated about plane global axes acting through n the current plane centroid as calculated from the average of all cut elements.</td></tr>
<tr><td>Fy is force in the global Y axis</td><td>Myy is moment about the global YY axis</td></tr>
<tr><td>Fz is force in the global Z axis</td><td>Mzz is moment about the global ZZ</td></tr>
</table>

Changing the coordinate system in which results are reported



Deformed space cut forces and moments can also be rotated to the global coordinate system for reporting puposes using the "System" popup menu.

This affects all reporting of forces and moments, both in the Cut sections panel and in **Write** and **XY_Plot**

Note that this is simply a geometric transformation of the coordinate system in which results are expressed, rotating them between global and plane local systems.

Although forces in the global system are reported as [Fx, Fy, Fz] they still represent a normal force and two shear forces in the original system of the plane. Forces on a plane are not the same as forces at a point in space!

In order to obtain "direct" (not shear) forces through the structure in all three global axes it will be necessary to create three cutting planes aligned with each of the global axes, and to collect results from each in turn.

Compatibility with *DATABASE_CROSS_SECTION output from LS-DYNA

LS-DYNA uses the lagrangian approach for cross-sections, and computes their forces using the equivalent of the BASIC method above. Results from LS-DYNA should match those from D3PLOT closely when BASIC space is used..

From D3PLOT 10.0 onwards, D3PLOT is capable of displaying any *DATABASE_CROSS_SECTION definitions in the input deck, and also extracting the forces reported by LS-DYNA in these. For this to work all of the following must be true:

- You must be running D3PLOT 10.0 or later
- It must have read a ZTF file generated by PRIMER 10.0 or higher (in order to determine the geometry)
- It must have read the "binout" file generated by LS-DYNA (in order to extract the cross section forces)
- Cross-section output *DATABASE_SECFORC must have been turned on, and binary output (to the binout file) turned on.

D3PLOT ***Cut*** sections, and LS-DYNA ***Cross*** sections are separate and different within D3PLOT:

- D3PLOT ***Cut*** sections, as described in this manual section,
    - Are user-defined and can be modified dynamically during post-processing.

  - Cut dynamically through the model using graphics calculations to display the cut structure.
  - Calculate forces and moments from a limited subset of elements using the forces, moments and stresses reported in those elements.
  - Force and Moment calculation can be in local or global systems, and the user can control dynamically (by blanking) the elements in which it takes place.
  - Only a single cut section can be active at a time, although any number may be stored for later retrieval.
- LS-DYNA ***Cross*** sections, as defined under *DATABASE_CROSS_SECTION in the LS-DYNA user manual:
  - Are defined in the original keyword input deck.
  - Have their forces and moments calculated by LS-DYNA during the analysis, and reported to ASCII secforc and/or binout files
  - The section geometry and elements which are cut are stipulated in the input deck, and cannot be changed during post-processing.
  - The force system of the results is always global, and the cutting space lagrangian (basic). Neither can be changed during post-processing.
  - Can have their geometry imported and displayed in D3PLOT via a ZTF file
  - Can have their results, which are always in the global system, extracted from a binout file and displayed in D3PLOT.
  - Any number of cross sections may be defined, and all can be displayed in D3PLOT if read as described above.

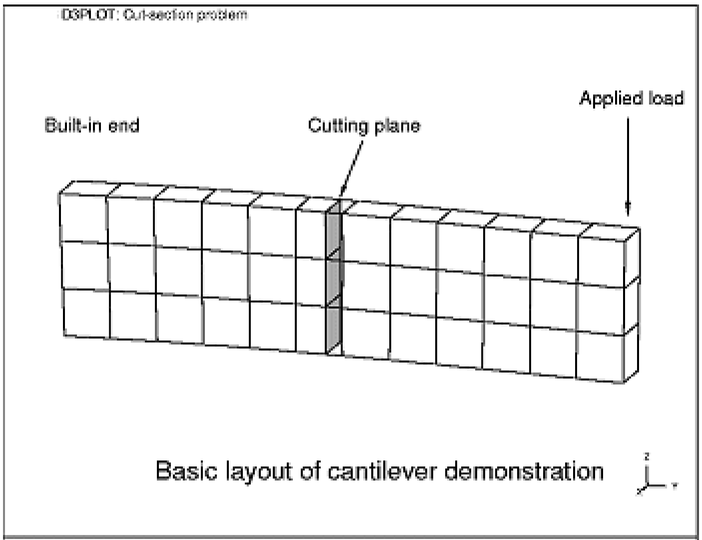
It is possible to use the geometry of an LS-DYNA Cross section definition to define a D3PLOT Cut section, which will overlay the two definitions. Selecting Basic space for display and the reporting of forces and moments should give very similar results. The results will probably not be identical for one or more of the following reasons:

- D3PLOT extracts results from the PTF files, and the time of a given state may not match exactly the nearest time written at "time history frequency" in the binout file. If forces are varying rapidly this can give rise to significant differences.

- D3PLOT can only work with the limited subset of element data available in the PTF file. (No bending data in thick shells, no spring output, etc)

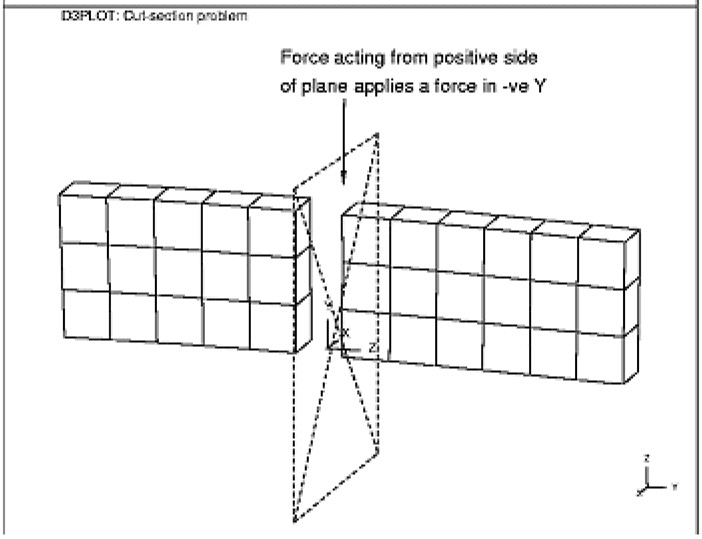
- D3PLOT cut sections calculate forces from all unblanked elements that are intersected, whereas LS-DYNA cross sections can control the elements considered both by defining a subset of parts and by limiting the extent of the section in its local XY plane. Careful blanking could be used to reproduce the same result, but it can be difficult to get exactly right.

Consider the following example of a cantilever cut along its length:



In this example a cantilever made of solids is loaded downwards at its free end.

There is a cutting plane defined in DEFORMED space roughly half way along its length, with the positive side (+ve Z axis) being the free end.

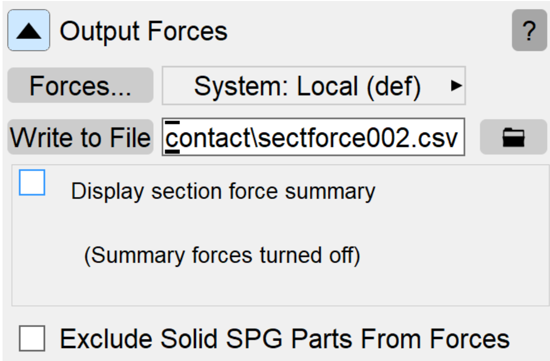


The force acting on the cut plane from its +ve side acts downwards.

In the plane local coordinate system this is in the negative Y sense.

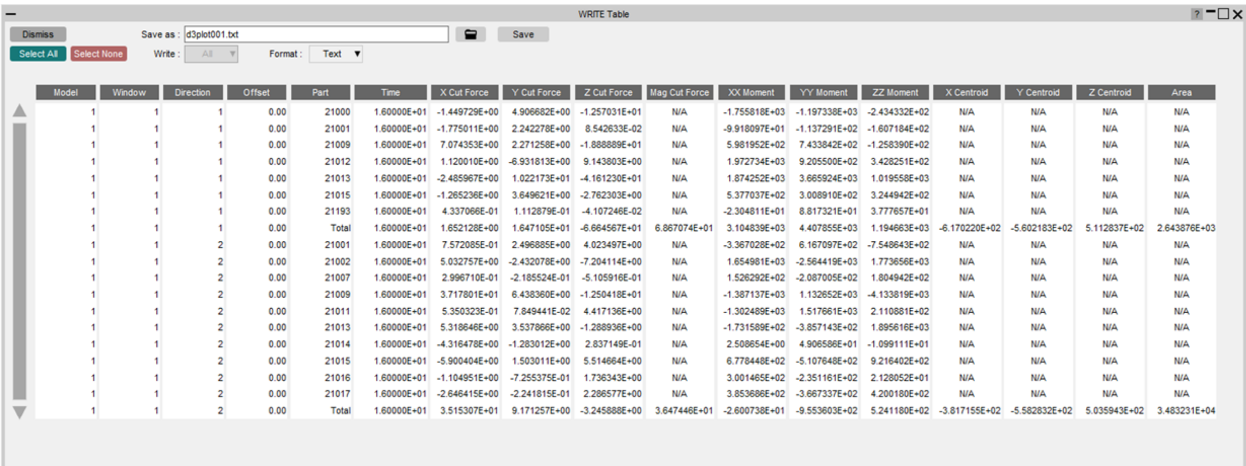
Methods of obtaining *written* force output

Method 1: Instantaneous force output on Cut-section menu



Method 2: **FORCES...** button in the Cut-sections menu

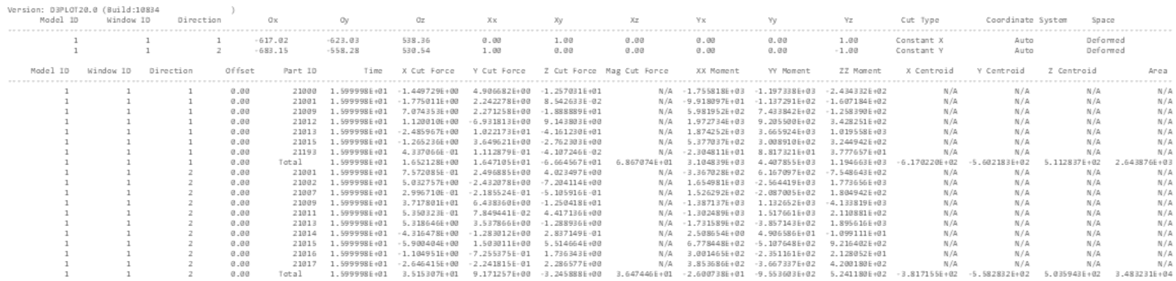
The table of forces given here has been improved in D3PLOT 20.0. It now uses the same style as the Write table. It contains a row for each combination of model ID, window ID, direction index, offset (to distinguish parallel planes) and part ID. Each row shows the contribution of that part to the cut section force and moment components. For each combination of model, window, direction and offset there is also a row for the total across all parts, which additionally shows the cut force magnitude, the coordinates of the centroid of the intersection of the plane, and the area. Remember that elements which are blanked are not included, and parts which do not contribute any force are also not reported in order to keep the list short. An example is shown below:



Model	Window	Direction	Offset	Part	Time	X Cut Force	Y Cut Force	Z Cut Force	Mag Cut Force	XX Moment	YY Moment	ZZ Moment	X Centroid	Y Centroid	Z Centroid	Area
1	1	1	0.00	21000	1.60000E+01	-1.449729E+00	4.906682E+00	-1.257031E+01	N/A	-1.755818E+03	-1.197338E+03	-2.434332E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21001	1.60000E+01	-1.775011E+00	2.242278E+00	8.542633E-02	N/A	-9.918097E+01	-1.137291E+02	-1.607184E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21009	1.60000E+01	7.074353E+00	2.271258E+00	-1.888889E+01	N/A	5.981952E+02	7.433842E+02	-1.258390E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21012	1.60000E+01	1.120010E+00	-6.931813E+00	9.143803E+00	N/A	1.972734E+03	9.205500E+02	3.428251E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21013	1.60000E+01	-2.485967E+00	1.022173E+01	-4.161230E+01	N/A	1.874252E+03	3.665924E+03	1.019558E+03	N/A	N/A	N/A	N/A
1	1	1	0.00	21015	1.60000E+01	-1.265236E+00	3.649621E+00	-2.762303E+00	N/A	5.377037E+02	3.008910E+02	3.244942E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21193	1.60000E+01	4.337066E-01	1.112879E-01	-4.107246E-02	N/A	-2.304811E+01	8.817321E+01	3.777657E+01	N/A	N/A	N/A	N/A
1	1	1	0.00	Total	1.60000E+01	1.652128E+00	1.647105E+01	-6.664567E+01	6.867074E+01	3.104839E+03	4.407855E+03	1.194663E+03	-6.170220E+02	-5.602183E+02	5.112837E+02	2.643876E+03
1	1	2	0.00	21001	1.60000E+01	7.572085E-01	2.496885E+00	4.023497E+00	N/A	-3.367028E+02	6.167097E+02	-7.548643E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21002	1.60000E+01	5.032757E+00	-2.432078E+00	-7.204114E+00	N/A	1.654981E+03	-2.564419E+03	1.773656E+03	N/A	N/A	N/A	N/A
1	1	2	0.00	21007	1.60000E+01	2.996710E-01	-2.185524E-01	-5.105916E-01	N/A	1.526292E+02	-2.087005E+02	1.804942E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21009	1.60000E+01	3.717801E+01	6.438360E+00	-1.250418E+01	N/A	-1.387137E+03	1.132652E+03	-4.133819E+03	N/A	N/A	N/A	N/A
1	1	2	0.00	21011	1.60000E+01	5.350323E-01	7.849441E-02	4.417136E+00	N/A	-1.302489E+03	1.517661E+03	2.110881E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21013	1.60000E+01	5.318646E+00	3.537866E+00	-1.288936E+00	N/A	-1.731589E+02	-3.857143E+02	1.895616E+03	N/A	N/A	N/A	N/A
1	1	2	0.00	21014	1.60000E+01	-4.316478E+00	-1.283012E+00	2.837149E-01	N/A	2.508654E+00	4.906586E+01	-1.099111E+01	N/A	N/A	N/A	N/A
1	1	2	0.00	21015	1.60000E+01	-5.900404E+00	1.503011E+00	5.514664E+00	N/A	6.778448E+02	-5.107648E+02	9.216402E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21016	1.60000E+01	-1.104951E+00	-7.255375E-01	1.736343E+00	N/A	3.001465E+02	-2.351161E+02	2.128052E+01	N/A	N/A	N/A	N/A
1	1	2	0.00	21017	1.60000E+01	-2.646415E+00	-2.241815E-01	2.286577E+00	N/A	3.853686E+02	-3.667337E+02	4.200180E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	Total	1.60000E+01	3.515307E+01	9.171257E+00	-3.245888E+00	3.647446E+01	-2.600738E+01	-9.553603E+02	5.241180E+02	-3.817155E+02	-5.582832E+02	5.035943E+02	3.483231E+04

Force data can be saved as Text, CSV and XLSX file from the top of this table.

The output files contain additional header information recording the properties of each cut at the top of the listing.



Version: D3PLOT20.0 (Build:10834)

Model ID	Window ID	Direction	Ox	Oy	Oz	Xx	Xy	Xz	Yx	Yy	Yz	Cut Type	Coordinate System	Space
1	1	1	-617.02	-623.03	538.36	0.00	1.00	0.00	0.00	0.00	1.00	Constant X	Auto	Deformed
1	1	2	-683.15	-558.28	530.54	1.00	0.00	0.00	0.00	0.00	-1.00	Constant Y	Auto	Deformed

Model ID	Window ID	Direction	Offset	Part ID	Time	X Cut Force	Y Cut Force	Z Cut Force	Mag Cut Force	XX Moment	YY Moment	ZZ Moment	X Centroid	Y Centroid	Z Centroid	Area
1	1	1	0.00	21000	1.599998E+01	-1.449729E+00	4.906682E+00	-1.257031E+01	N/A	-1.755818E+03	-1.197338E+03	-2.434332E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21001	1.599998E+01	-1.775011E+00	2.242278E+00	8.542633E-02	N/A	-9.918097E+01	-1.137291E+02	-1.607184E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21009	1.599998E+01	7.074353E+00	2.271258E+00	-1.888889E+01	N/A	5.981952E+02	7.433842E+02	-1.258390E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21012	1.599998E+01	1.120010E+00	-6.931813E+00	9.143803E+00	N/A	1.972734E+03	9.205500E+02	3.428251E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21013	1.599998E+01	-2.485967E+00	1.022173E+01	-4.161230E+01	N/A	1.874252E+03	3.665924E+03	1.019558E+03	N/A	N/A	N/A	N/A
1	1	1	0.00	21015	1.599998E+01	-1.265236E+00	3.649621E+00	-2.762303E+00	N/A	5.377037E+02	3.008910E+02	3.244942E+02	N/A	N/A	N/A	N/A
1	1	1	0.00	21193	1.599998E+01	4.337066E-01	1.112879E-01	-4.107246E-02	N/A	-2.304811E+01	8.817321E+01	3.777657E+01	N/A	N/A	N/A	N/A
1	1	1	0.00	Total	1.599998E+01	1.652128E+00	1.647105E+01	-6.664567E+01	6.867074E+01	3.104839E+03	4.407855E+03	1.194663E+03	-6.170220E+02	-5.602183E+02	5.112837E+02	2.643876E+03
1	1	2	0.00	21001	1.599998E+01	7.572085E-01	2.496885E+00	4.023497E+00	N/A	-3.367028E+02	6.167097E+02	-7.548643E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21002	1.599998E+01	5.032757E+00	-2.432078E+00	-7.204114E+00	N/A	1.654981E+03	-2.564419E+03	1.773656E+03	N/A	N/A	N/A	N/A
1	1	2	0.00	21007	1.599998E+01	2.996710E-01	-2.185524E-01	-5.105916E-01	N/A	1.526292E+02	-2.087005E+02	1.804942E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21009	1.599998E+01	3.717801E+01	6.438360E+00	-1.250418E+01	N/A	-1.387137E+03	1.132652E+03	-4.133819E+03	N/A	N/A	N/A	N/A
1	1	2	0.00	21011	1.599998E+01	5.350323E-01	7.849441E-02	4.417136E+00	N/A	-1.302489E+03	1.517661E+03	2.110881E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21013	1.599998E+01	5.318646E+00	3.537866E+00	-1.288936E+00	N/A	-1.731589E+02	-3.857143E+02	1.895616E+03	N/A	N/A	N/A	N/A
1	1	2	0.00	21014	1.599998E+01	-4.316478E+00	-1.283012E+00	2.837149E-01	N/A	2.508654E+00	4.906586E+01	-1.099111E+01	N/A	N/A	N/A	N/A
1	1	2	0.00	21015	1.599998E+01	-5.900404E+00	1.503011E+00	5.514664E+00	N/A	6.778448E+02	-5.107648E+02	9.216402E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	21016	1.599998E+01	-1.104951E+00	-7.255375E-01	1.736343E+00	N/A	3.001465E+02	-2.351161E+02	2.128052E+01	N/A	N/A	N/A	N/A
1	1	2	0.00	21017	1.599998E+01	-2.646415E+00	-2.241815E-01	2.286577E+00	N/A	3.853686E+02	-3.667337E+02	4.200180E+02	N/A	N/A	N/A	N/A
1	1	2	0.00	Total	1.599998E+01	3.515307E+01	9.171257E+00	-3.245888E+00	3.647446E+01	-2.600738E+01	-9.553603E+02	5.241180E+02	-3.817155E+02	-5.582832E+02	5.035943E+02	3.483231E+04

Method 3: **WRITE TO FILE** button in the Cut-sections menu

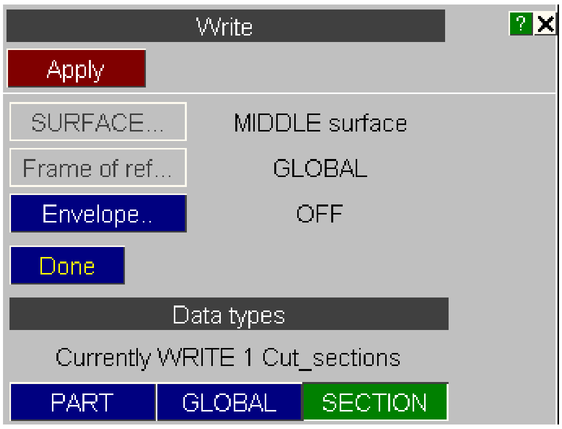
Outputs the same data as above, but writes it to a CSV file

Method 4: **WRITE** Output (as an entity type)

When a cut-section is current the **SECTION** entity will be available as an "entity" type in the **WRITE** menu:

To use this select **SECTION** and then a data component.

You can select a scalar component, (such as **FX_X_CUT_FORCE** as shown here), or a summary of all forces and moments.



The advantage of using **WRITE** is that the results can be directed to file for subsequent use. An example of such output is given below

++++++++ Data at time .30046E-02 ++++++++						
CUT_SECTIONs:	listing of CS_CUT_SUMMARY					
Cut section:	Fx	Fy	Fz	Mxx	Myy	Mzz
--						

--						
Section	-9.667E+03	1.290E-04	-2.004E+00	-5.770E+01	-1.114E+06	1.1166E+05

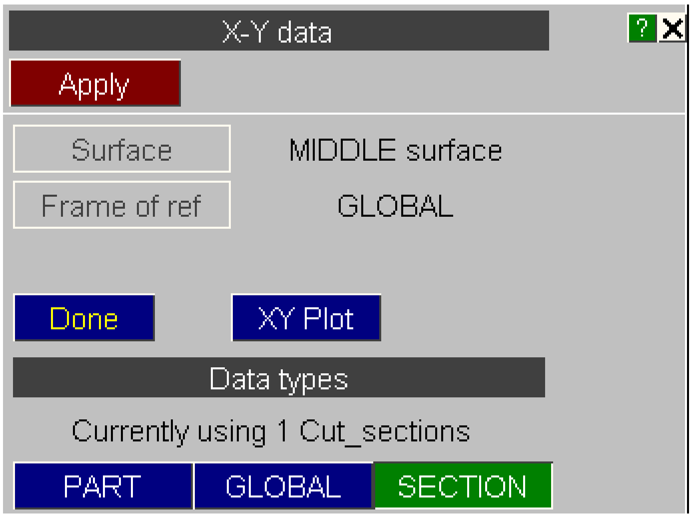
Obtaining XY graphs of cut-section forces wrt. time.

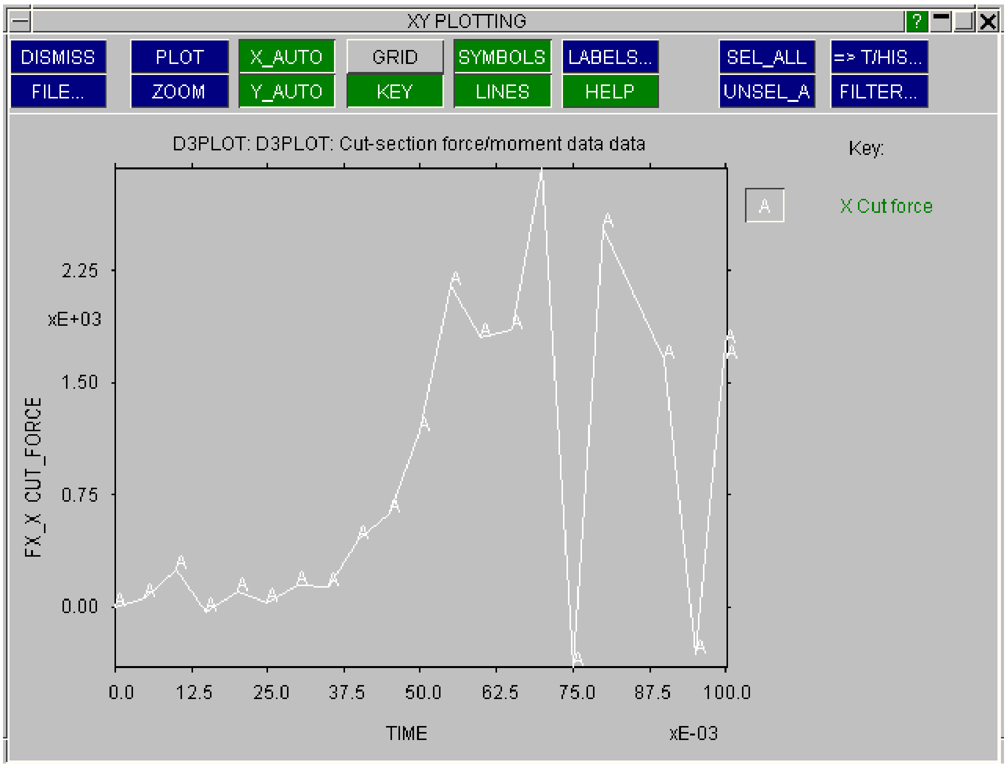
It is possible to use cut-section forces as a valid entity type for output in the **XY_PLOT** command, in a fashion similar to **WRITE** . When there is a current cut-section the entity type **SECTION** will be available in the **XY_PLOT** panel.

Choose the **SECTION** "entity" type, followed by a scalar data component, and the results of that component versus time will be generated.

Results can be written to disk as "curve" files, using the name `sect001.cur` etc, and also drawn as graphs in D3PLOT.

The figure below shows typical output of X cut force vs. time.





Note that you can also write plotted results to a curve file using the **FILE...** command on the plotting panel. (See [Section 6.8](#) for more details.)

9.5.13.5. Cut Area and Centroid

Cut area and centroid

It is also possible to extract data components **CUT_AREA** and **CX** / **Y** / **Z_CENTROID** of the cut section for display in **WRITE** and **XY_DATA** .

CUT_AREA Is the sum of all **Solid** , **Shell** and **Thick shell** cut face areas. No attempt is made to compute the area of cuts through beams - see [the notes above](#) on computing cut centroids for beams..

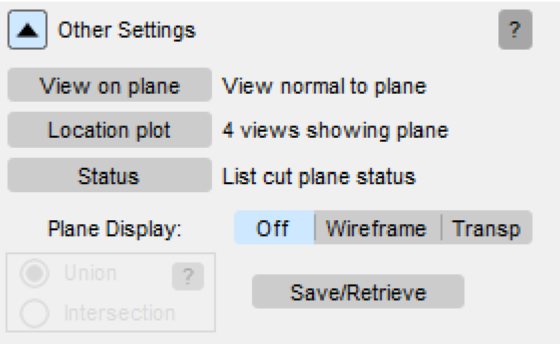
CX/Y/Z_CENTROID Is the average coordinates of the cut face, computed from the 1st moment of area of the cut elements areas. Therefore it suffers from the limitation above in that it cannot include both solids/shells and beam elements.

- Where a model mixes solids &/or shells and beams then only the centroid through the solids &/or shells is reported.
- If only beam elements are cut then the centroid of the cut is computed assuming that each beam has an equal area.

Note that this is the case for both **Basic** and **Deformed** space sections, and in the latter case the reported centroid may be very different to the section origin (although it will be on the section plane).

9.5.14. OTHER Cut Section Options

OTHER Cut-Section Options



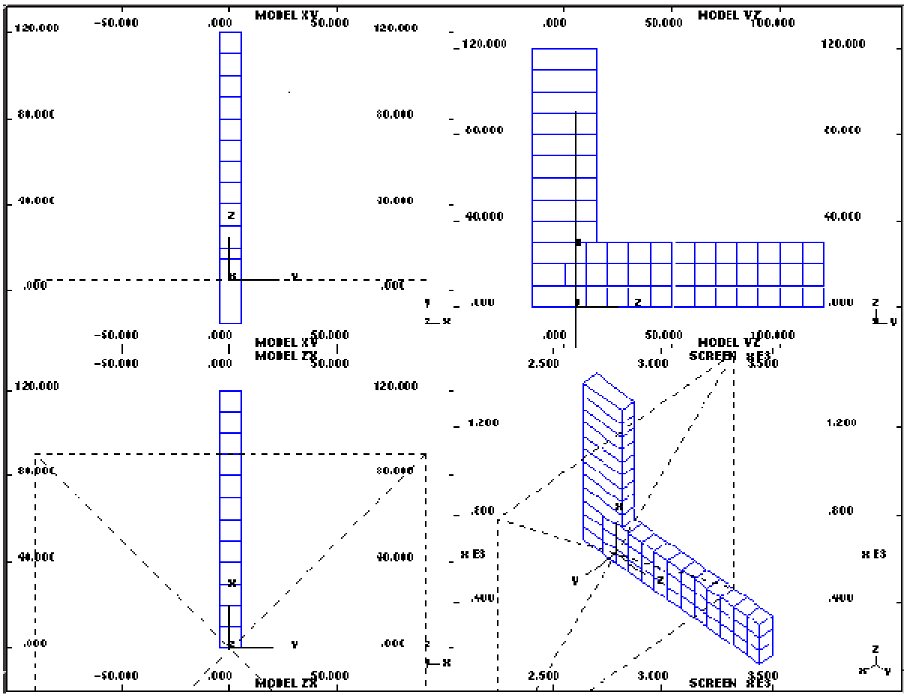
VIEW_PLANE Sets the current view to be normal to the plane

The current view will be set so that the cutting plane X, Y, Z axes are aligned with screen X, Y, Z; giving a view exactly normal to the plane. The image is only rotated, not scaled, so it may be necessary to autoscale or apply a translation to get the view you want.

LOC_PLOT Draws a "location" plot

This shows three standard views down each of the model X, Y and Z axes; and also the current view. The plane position is sketched on each image.

A "Graticule" is drawn on each view to provide dimensions. (Note that this image is not stored in laser files.)



9.5.14.1. SAVE/RETRIEVE Managing the Storage and Retrieval of Cut Section Definitions on Disk

SAVE/RETRIEVE Managing the storage and retrieval of cut-section definitions on disk

There is only ever one "current" cutting-plane definition, but up to 100 such definitions can be stored in an external " `section.cut` " file, and any number of such files may exist.

Sections are model-independent and may be shared between dissimilar analyses.

Storing and retrieving cut-sections :

This figure shows the storage and retrieval sub-menu. The four commands in the left hand column manipulate sections as follows:

STORE	Stores the current section definition in the file.
GET	Reads a stored definition which overwrites the current one.
RENAME	Renames a stored definition.
DELETE	Deletes a stored definition.
FILE...	Lets you enter a new " `section.cut` " filename: Any filename is permissible, but `section.cut` is assumed, and the extension " `.cut` " is recommended (but not mandatory). Note that `section.cut` files are binary, and are not normally

transferrable between different machine types. Nor will you be able to read or edit them.
However transfers between typical workstations (using IEEE format) will usually work OK.

Only **GET** affects the current definition, the other commands leave it unchanged.

All storage and retrieval operations take place using the current " `section.cut` " file. If such a file has not been opened explicitly a file called section.cut is opened automatically (and an empty file of this name is created if it doesn't already exist.)

You will note that sections are stored with names as well as numbers. These are optional, but help when identifying which section does what. A maximum of 40 characters is permitted for each section name.

9.5.15. Using Cut Sections Under OpenGL in 3D Mode

Using cut sections under OpenGL in 3D mode

D3PLOT has always displayed deformed space (Eulerian) cut sections correctly because they are flat and OpenGL provides "clipping planes" that will intersect the model correctly. D3PLOT still has to synthesise the cut surfaces of solids and capping polygons for shells, but the hardware has looked after the rest of the problem.

However prior to D3PLOT 10.0 basic space (Lagrangian) planes, which cease to be flat once the model deforms, have not been rendered correctly in OpenGL 3D mode. The cut surfaces through solids and the capping polygons for shells were displayed correctly, but the intersection of the rest of the model with the plane worked on the deformed space (flat) plane in 3D mode, which was in the wrong place, and gave a "jagged edge" effect in 2D mode.

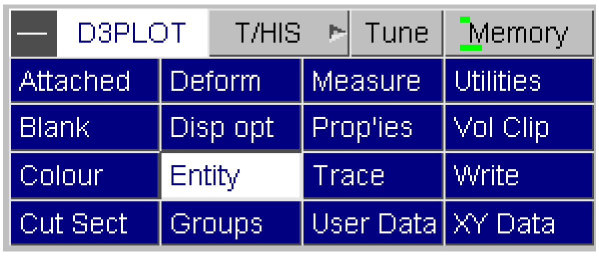
From D3PLOT 10.0 onwards this limitation has been removed, and basic space cut sections now render correctly in 3D graphics mode.

The intersection between the plane and elements with a determinate topology (solids, shells, thick shells, beams, springs, seatbelts) is calculated individually for each item, and the fraction of the symbol which should be visible on the "drawn" side of the plane is shown correctly.

Nodes themselves, and elements with only a single node (SPH elements, masses, spotwelds, etc), are treated as being either "wholly visible" or "wholly clipped" by the plane and are drawn in full or not drawn accordingly. This logic is also applied to items without a determinate topology (eg joints, rigidwalls) and to composite elements such a spotweld clusters. This means that their symbols, which are of finite size, will pop into and out of view as the plane crosses them rather than being intersected gradually. This is not strictly correct, but it is acceptable and - more to the point - is fast to process and gives good graphics speeds.

9.6. ENTITY Switching the Display of Entity Categories On/Off

ENTITY Switching the display of entity categories on/off



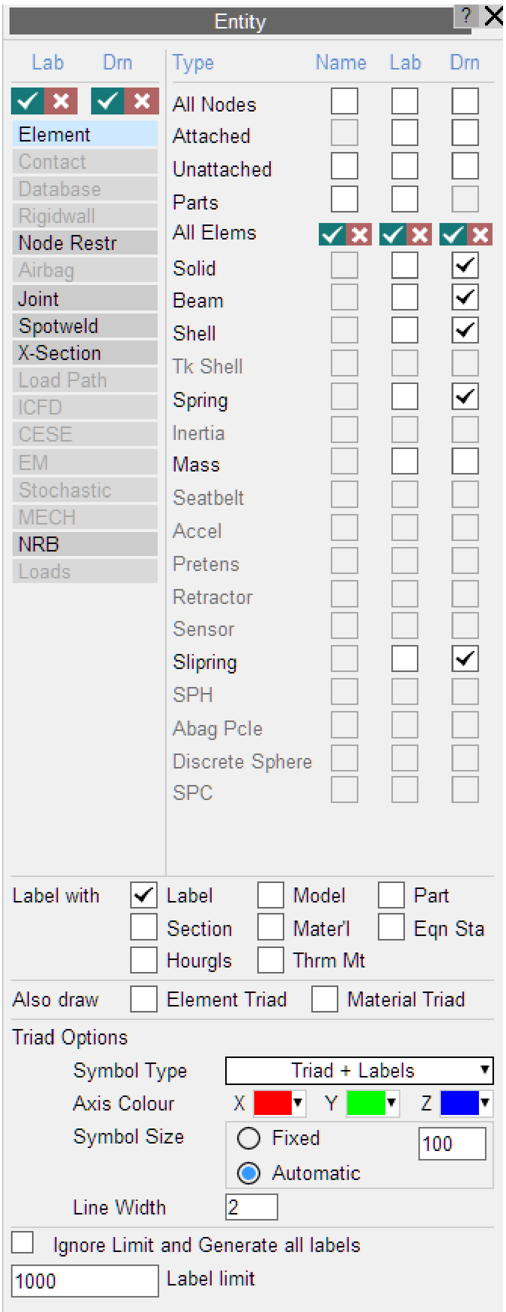
You can use **BLANKING** to control the visibility of selected nodes and elements, but sometimes you may wish to remove a complete category of entity type from the display list. The **ENTITY** menu provides this capability: when an entity category is switched off it is never displayed in subsequent plots (of any type) until explicitly turned back on again.

The entity panel also controls label display and the display of entity names. By default only the elements in a model are drawn, with no labels, node symbols or other information appended to them.

You can add extra information to plots, control the display of classes of information and label items dynamically on the screen using the menu. This can be accessed in 3 ways, the shortcut key E, the top bar menu **DISPLAY** > **ENTITIES** or the button **ENT** . This panel controls the display of elements and nodes, (i.e. basic "structural" items); also their symbols, labels and local direction triads as well as the display of "other" items, such as constraints, contacts, rigidwalls, etc; and also their labels, symbols and other related displayable data.

It must be stressed that these commands only permit or deny the display of *classes* of information, they do not control the visibility of individual items or models.

For example they might be used to enable the display of nodes. This would permit nodes in any models to be displayed provided they were not made invisible by some other command.



The left hand column of the panel dictates the display of the right hand column. At any one time a "master" category will be selected from the left-hand column (in this example **Elements** is selected). The "master" categories each contain further "child" categories below them. The right hand column displays the appropriate "child" categories for the selected "master".

The **Label** columns control whether or not the items will be labelled (with the information selected under **Label with.** The **Drawn** columns control whether or not the items will be drawn. "Child"categories can be controlled individually (in the example shown the display of beams has been turned off), however they cannot be set so as to be drawn or labelled if the "master" category is not set to be (setting a "child" to be on will automatically switch the "master" setting on).

Some LS-DYNA entity types can have a name defined for them. The **Name** columns control which items will have their names displayed if they are defined. Both the label and name of an entity can be displayed at the same time.

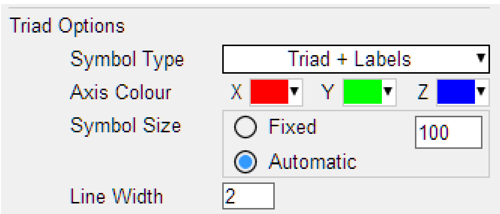
Label with determines what is actually drawn as a "label" when labelling is selected for an element or node class.

Selecting multiple labelling categories will lead to compound labels being generated (eg **M1/H1001/P12/**).

As this example illustrates different models may contain different entity types. Where all currently selected models contain a type (for example Shells here) the selection box has a white background, whereas if only a subset do then the box will have a grey background to denote its "mixed" status.

Triad Options

From version 21 onwards the display of element and material triads has been speeded up and a number of new options have been added to control how the triads are displayed.



These options include

- Triad + X, Y, Z labels
- Coloured Triad (x axis Red, y axis Green, z axis Blue)
- X Axis Only
- Y Axis Only
- Z Axis Only

The default option is to display the complete Triad with X, Y and Z labels.

New options have also been added that control the line width and the size of the triad symbols.

The following preferences can be used to change the default Triad properties.

d3plot*triad_mode	Controls the display mode for Element and Material Triad Symbols. This can be set to : TRIAD+LABELS, COLOURED_TRIAD, X-AXS_ONLY, Y-AXS_ONLY, Z-AXS_ONLY
d3plot*triad_size	Display triads using either a FIXED size or an AUTOMATIC size
d3plot*triad_fixed_size	Size used for FIXED size triads
d3plot*triad_line_width	Line width in pixels used for drawing triads
d3plot*triad_x_axis_colour	Colour of X axis when “triad_mode” is set to “COLOURED_TRIAD”
d3plot*triad_y_axis_colour	Colour of Y axis when “triad_mode” is set to “COLOURED_TRIAD”
d3plot*triad_z_axis_colour	Colour of Z axis when “triad_mode” is set to “COLOURED_TRIAD”

Controlling the number of labels displayed

Displaying a large number of labels can be very slow and plots will become very cluttered if too much information is displayed. By default D3PLOT will generate a warning if the labelling options selected generate more than 1000 labels.

The on screen warning is only displayed the 1st time this label limit is reached and the choice to either ignore the limit or to change the limit will be given.

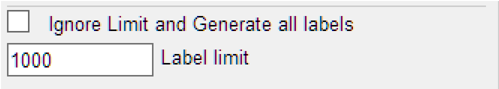
The default number of labels that D3PLOT displays before generating a warning can be changed used the preference option

```
d3plot*max_labels:   1000
```

The on screen warning message can also be disabled by setting the following preference to FALSE.

```
d3plot*label_warning:  FALSE
```

At any time the label limit can be reset or modified using the options in the entity panel.



9.6.1. Elements and Nodes (Structural Items)

Elements and nodes (Structural Items)

Nodes are a special case.

The display of nodes in D3PLOT is treated differently to that of elements: by default they are not drawn, but they are still "there" on a plot for the purposes of labelling and screen-picking.

This situation arises because it is convenient to be able to screen-pick and label nodes, but not useful to draw their symbols: that would just slow down graphics and clutter up the screen. Therefore the treatment of nodes is as follows:

- Nodes on visible elements are always "there", even if not drawn explicitly, and may be screen-picked and labelled (by either method) at will.
- Turning the **Attached nodes** switch here **on** will cause those nodes "attached" to visible elements only to be displayed using "star" symbols.
- Turning the **All nodes** switch here **on** will cause all nodes, whether attached to an element (visible or otherwise) or not, to be displayed

Each class of element may be selected for display and/or labelling individually. Here the display of beams has been turned off.

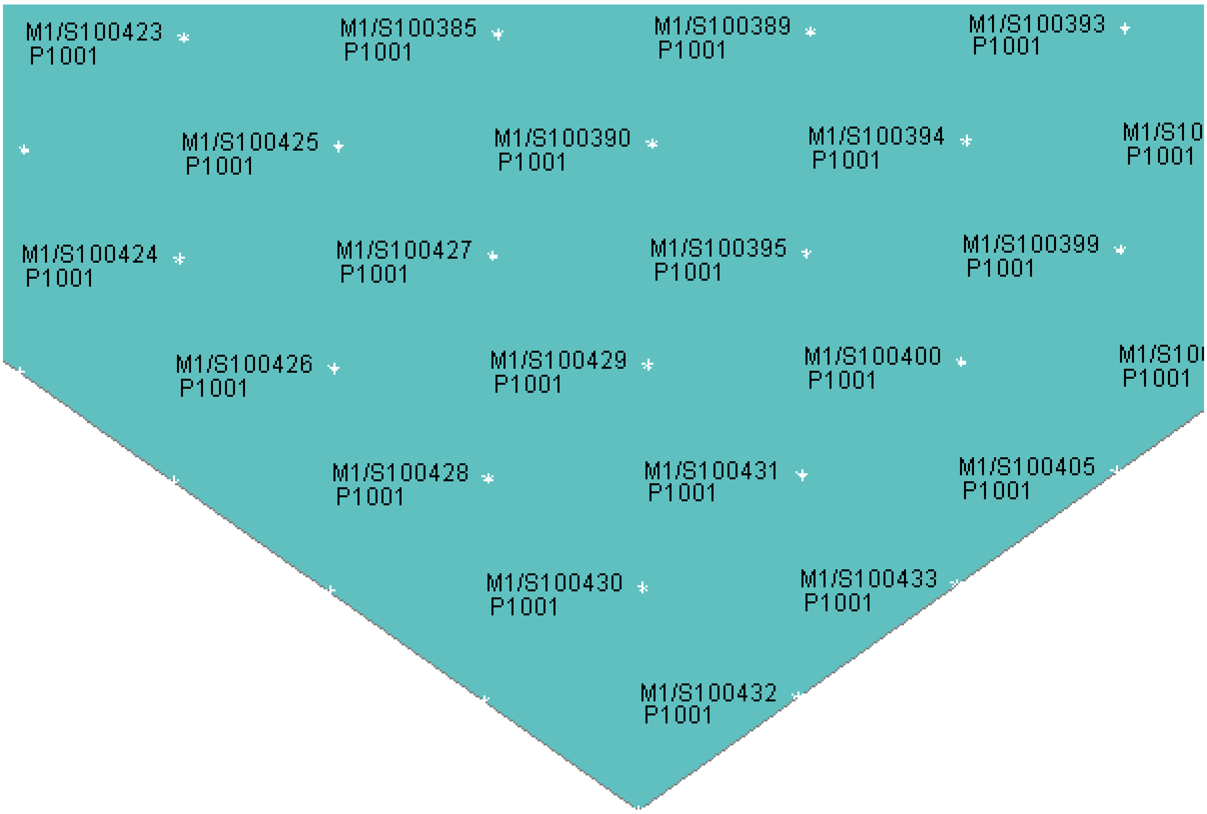
The **ALL ELEMS** row selects them all.

Associated data: Local direction **TRIADS** are drawn for element types with coordinate systems.

9.6.2. How Labelling on Plots is Handled for Nodes and Elements

How labelling on plots is handled for nodes and elements

Static labels are handled via the Entity panel (as described in this section). "Dynamic" labels may be added using Quick-Pick (see "QUICK PICK" Options)The default label is a node or element number, but a variable amount of information can be generated to form a "label" which can run to multiple lines, as this example shows.



This figure shows an example of shells which have been labelled with:

MODEL Mnnn for ***M*** odel number <nnn> **LABEL Snnn** for ***S*** hell <nnn>. **PART Pnnn** for ***P*** art <nnn>.
D3PLOT attempts to group labels logically and to locate them so that they don't overlap, but if you try to add too much information you will end up with a total mess on the page.

The "attached" nodes in this figure have also been switched on: these are drawn as asterisks (*) at the relevant element vertices.

9.6.3. Res/Constraints Displaying Nodal Restraints and Constraints

Res/Constraints Displaying nodal Restraints and Constraints

If a **ZTF** file is present, then restraint and constraint data will be available for any combination of nodes on:

***BOUNDARY**	**_SPC**	
	_PRESCRIBED_MOTION	
	_CYCLIC	
	_SLIDING	
***CONTACT**	**_CONSTRAINED_xxx**	
	_TIED_xxx	
***CONSTRAINED**	**_LINEAR**	
	_WELD_xxx	
	_NODE_SET	
	_NODAL_RIGID_BODY	
***PART**	which is rigid	

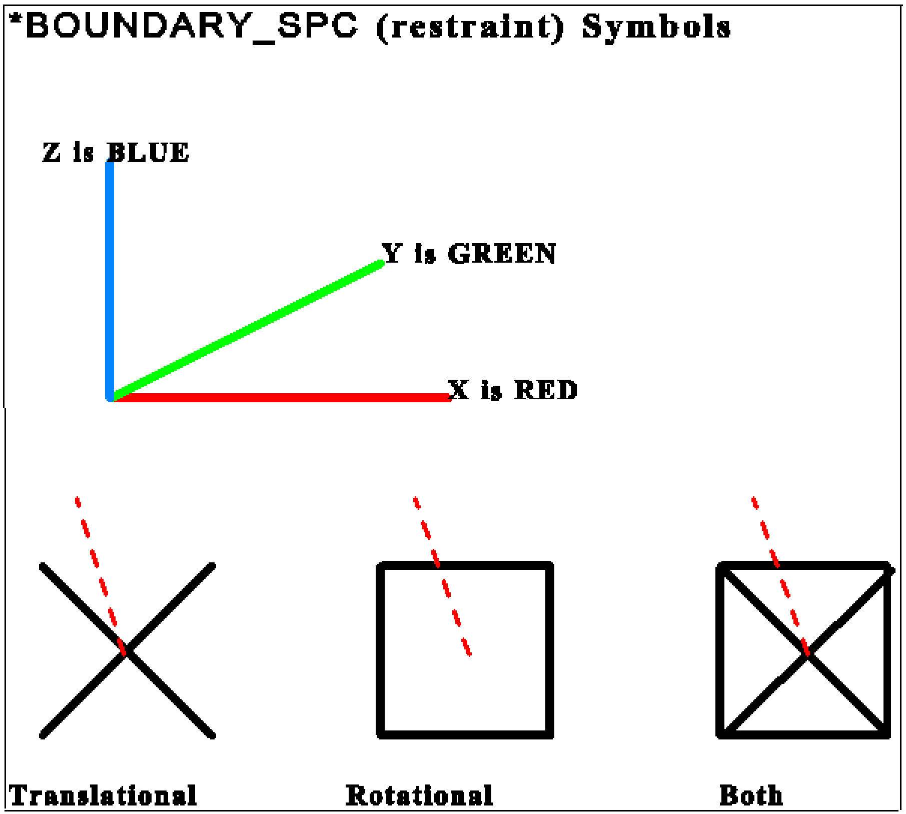
The image below shows how symbols for translational and rotational restraints and constraints are displayed.

X direction : Red
Y direction : Green
Z direction : Blue

Translational restraints have a ***cross*** at their end
Rotational restraints have a ***box*** at their end.

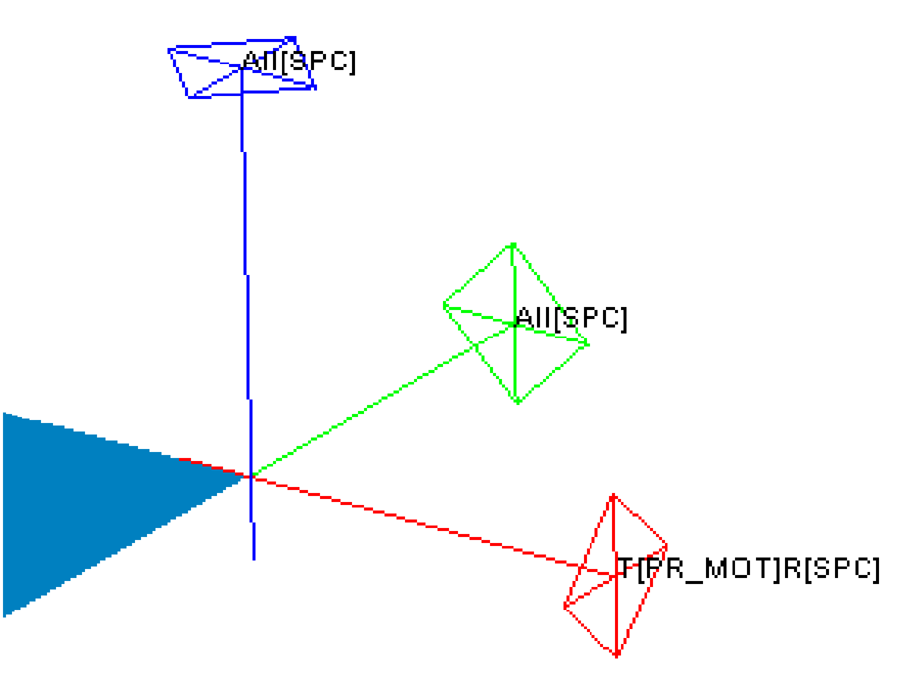
Turning on "labels" for the relevant item will give a brief description of the source of the re/constraint.

"T" designates translational restraint
"R" designates rotational restraint
"All" designates both translational and rotational



Here is an example of SPCs and Prescribed motion re/constraints from an actual model.

This image has "labels" turned on, and has been enlarged to show the symbols more clearly. Note that in the X direction translational constraint is due to a Boundary_Prescribed_Motion, whereas rotational restraint has been applied via a Boundary_SPC.



9.6.4. Spotwelds

Spotwelds

Spotweld solids, beams and clusters are a special case as they are solid and beam elements which can also have spotweld data components plotted on them.

When plotting these spotweld types the user can select how the spotwelds elements are treated and what results are displayed on them..

Never draw as structural elements

With this option elements that are spotweld beams and solids are always treated as though they are spotwelds and are ignored for things like contouring and min-max values of any non spotweld data component. If for example you do a SI plot of Von-Mises stress then any spotweld solids will be drawn as uncontoured.

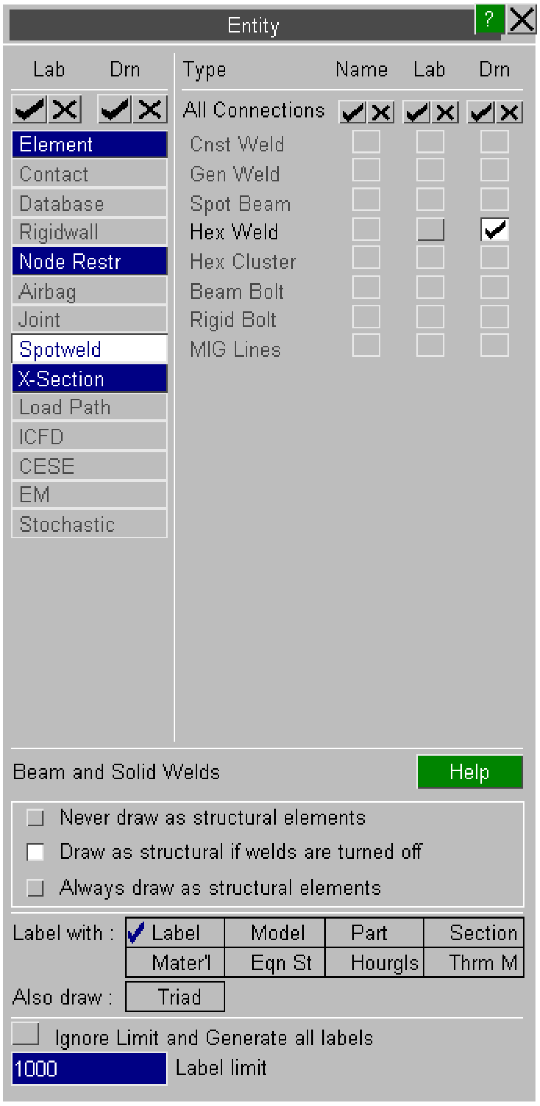
Draw as structural if welds are turned off.

With this setting spotweld beams and solids are treated as spotwelds and are ignored for things like contouring and min-max values if the entity switch for spotwelds is turned on.

If for example you do a SI plot of Von-Mises stress and the entity switch for spotweld solids is on then any spotweld solids will be drawn as uncontoured. If the switch is turned off then the spotwelds solids will be treated as normal elements and contoured.

Always draw as structural elements.

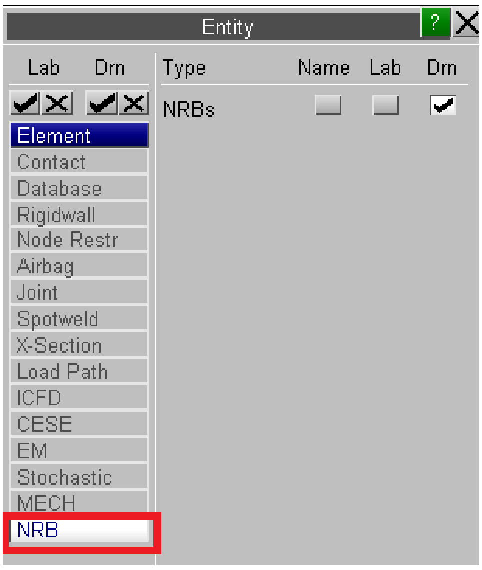
This is the opposite of (1). With this option the spotweld beams and solids are always treated as though they are normal structural elements and will be included in data plots for structural data components. If a spotweld data component is plotted then the spotweld beams and solids will be uncontoured but any generalized or constrained welds will still be contoured as normal.



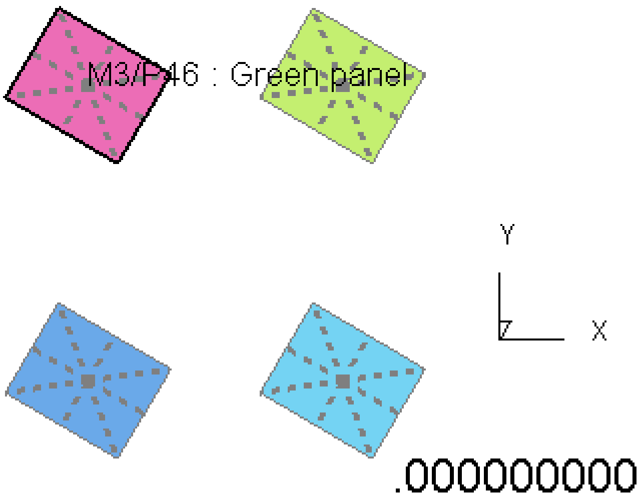
9.6.5. Nodal Rigid Bodies

Nodal Rigid Bodies

By default NRBs are drawn if the information has been transferred via the ztf file



These will be plotted as "spiders" joining a set of nodes to the master (first defined) node

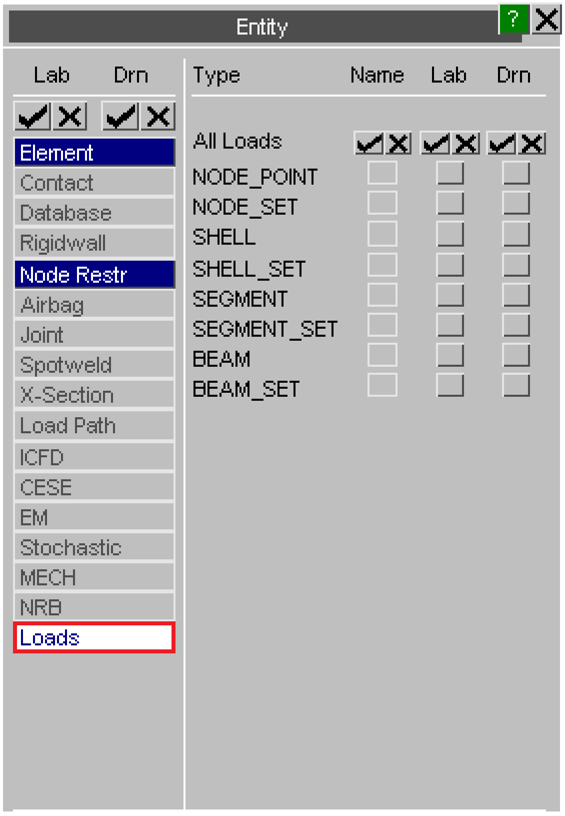


9.6.6. Loads

Loads

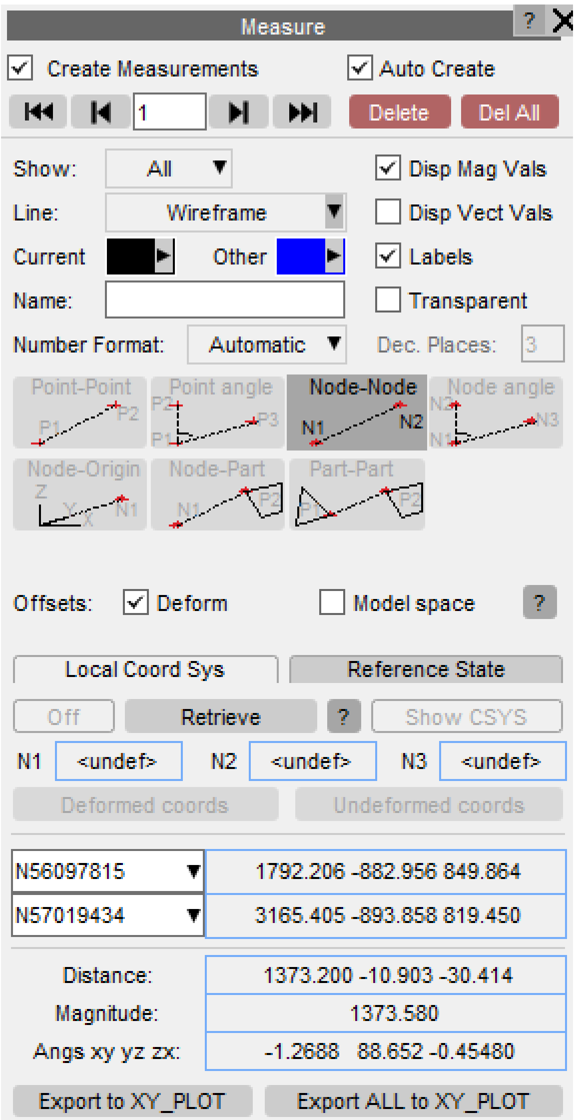
Loads can be drawn if the information has been transferred via the ZTF file. Currently the drawing of loads has the following limitations:

- Only the following types of loads can be drawn:
  - LOAD_NODE_POINT / SET,
  - LOAD_BEAM(_SET),
  - LOAD_SHELL(_SET),
  - LOAD_SEGMENT(_SET).



9.7. MEASURE Measuring Distances from the Screen

MEASURE Measuring distances from the screen



The figure (right) shows the **MEASURE** control panel (which can be accessed by pressing M as well as from the "Tools" menu).

From D3PLOT 10.0 onwards D3PLOT can keep track of up to 100 measurements. Each measurement can be in a different window or it can be between different models within the same window.


Select the first measurement


Select the previous measurement

Goto measurement (n)


Select the next measurement


Select the highest measurement defined + 1.

Once a measurement has been defined it will be drawn on the screen along with the corresponding value. By default all measurements will be drawn along with the values.

If this option is selected then the current measurement will be incremented as soon as enough points/nodes/parts have been selected. The preference *measure_auto_create* can be set to toggle the initial status of Auto Create.

Show All measurements, the Current measurement, or None.

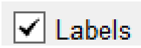
Draw the measurement lines in Wireframe mode (so they are always visible), or Hidden mode (so they are within the 3D model).

Specifies the colour used to draw the current measurement.

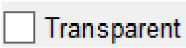
Specifies the colour used to draw all measurements apart from the current measurement.

Specifies a name to display next to the measurement.

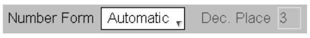
Display magnitude and vector values next to each measurement.



This option will add labels to any nodes or parts used to define the measurement.



This option will remove the square of background colour from behind the name, magnitude, and vector annotation on the plot.



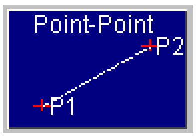
Specifies the format of the magnitude and vector values displayed on the plot.

The seven measurement functions here are described in below.

All of them use screen-picking with the cursor to select points, nodes or parts and report results in a special table in this panel.

"Point" functions:

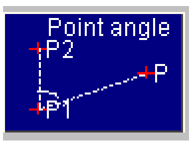
Screen points are arbitrary locations on the screen picked with the cursor. They are not located on or attached to the model in any way. Distances reported for screen points are reported in screen space units.



Point to Point **Measuring the distance between 2 points**

Pick pairs of points with the cursor, and the distance **in screen space units** between them is reported.

(Screen space units have the scale of model space, but projected onto the 2D plane of the screen.)

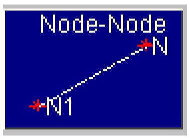


Point angle **Measuring the angle between 3 points**

Pick three points with the cursor. The angle (on the 2D screen plane) between vectors P1P2 and P1P3 is reported.

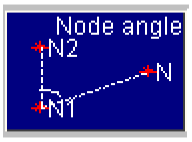
Node functions:

Nodes are picked on the model, and all distances and coordinates are reported in model space units. The nodal coordinates used are those of the state currently in core.



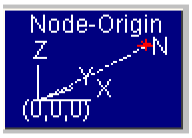
Node to Node **Measuring the distance between 2 nodes**

Pick two nodes with the cursor. The distance **in model space units** between them on XY, YZ and ZX planes and in 3D space is reported.



Node angle **Measuring the angle between 3 nodes**

Pick three nodes with the cursor. The angles on XY, YZ, ZX planes and in 3D space between vectors N1N2 and N1N3 are reported.

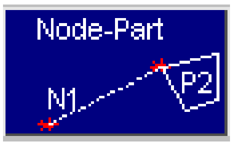


Node to Origin **Nodal position and distance from [0,0,0]**

Pick a node with the cursor. Its current coordinates and distance from the origin **in model space units** are reported.

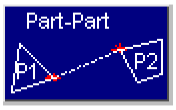
Part functions:

Parts are picked on the model (limited to SOLID, TSHELL and SHELL parts), and all distances and coordinates are reported in model space units. The coordinates used are those of the state currently in core.



Node to Part **Shortest distance between a node and a part**

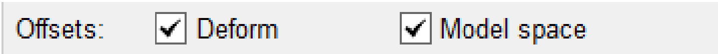
Pick a node and a part with the cursor. The shortest distance **in model space units** between them on XY, YZ and ZX planes and in 3D space is reported.



Part to Part **Shortest distance between two parts**

Pick two parts with the cursor. The shortest distance **in model space units** between them on XY, YZ and ZX planes and in 3D space is reported.

Offsets



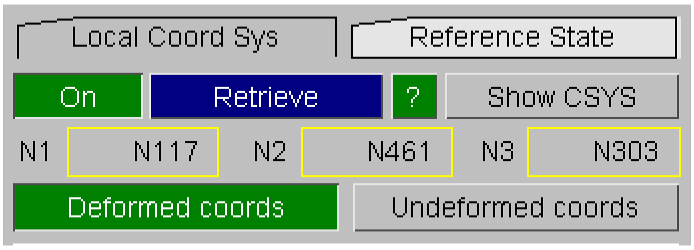
From D3PLOT 20.0 onwards measurements consider any offsets by Deform (for example Shift deform) or in model space for the coordinates in measure calculations. When these **Deform** and **Model space** options are turned off, measures are calculated from the actual coordinates of the nodes.

Local Coord Sys

Measurements can be made within a local coordinate system if desired. Press the **RETRIEVE** button to select a coordinate system from either the 'csys.loc' file (defined in the Shift Deformed panel) or any *DEFINE_COORDINATE definitions in the ZTF file.

Deformed coords / Undeformed coords

If a coordinate system defined by nodes is selected (i.e. from the 'csys.loc' file or a *DEFINE_COORDINATE_NODES definition in the ZTF file), you will have the option to use either the deformed or undeformed coordinates of the the nodes. If you select deformed the coordinate system will update with the model.



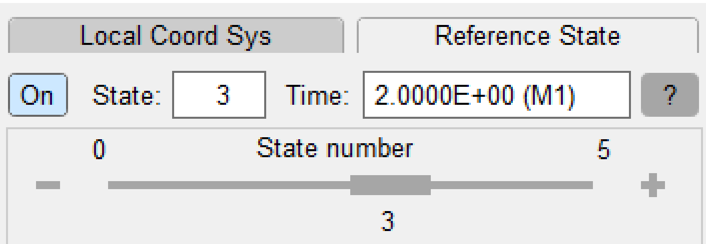
Reference State

Measurements can be relative to the value at a chosen reference state. Select the reference state by entering a value in the "State:" or "Time:" text boxes, or by moving the slider. If the "Time:" box is used, the closest state to that time will be selected.

Relative magnitudes of distances and angles are calculated by subtracting the reference state values from the current values using simple scalar subtraction. Below are the equations used in the calculations:

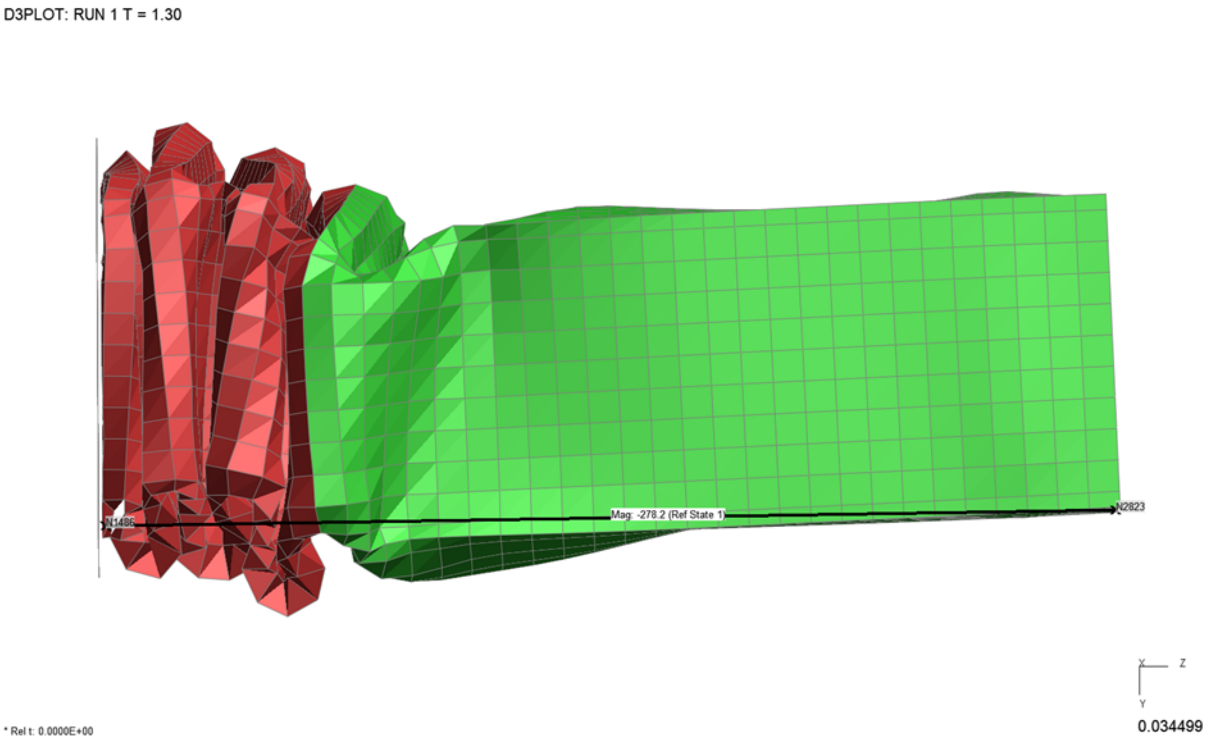
- Relative Displacement Magnitude = (Magnitude at current state) - (Magnitude at reference state)
- Relative XY Angle = (XY angle at the current state) - (XY angle at the reference state)
- Relative YZ Angle = (YZ angle at the current state) - (YZ angle at the reference state)
- Relative ZX Angle = (ZX angle at the current state) - (ZX angle at the reference state)

The X, Y, and Z displacement vectors displayed are unaffected when the reference state is active.

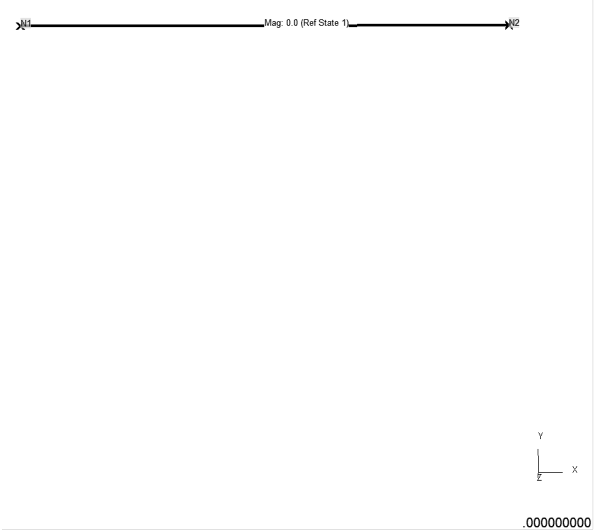


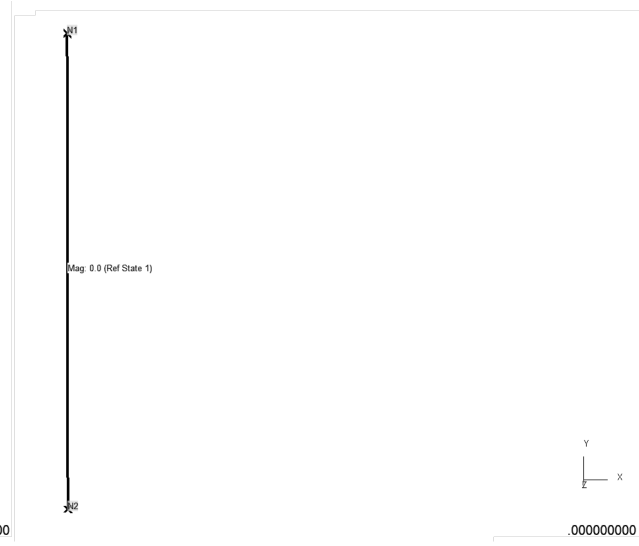
Typically, relative magnitude measurements would be useful in the case where the measurement is done for uniaxial load cases. An example is shown in the first model shown below where a crush tube is under compression. By setting the reference state to state 1, the user can immediately tell that a compression of about 278.2 units has occurred.

D3PLOT



There are situations where measuring relative to another state isn't very meaningful. This is illustrated in the second model. For example, if you measure the distance between two nodes and one rotates 90 degrees around the other, the relative distance remains zero. In the images below, the first image shows the beam in a horizontal position (state 1, the reference state), and the second image shows the beam rotated to a vertical position (state 2).





The vector distance along vector N1 to N2 in both current and reference states are 10 units, so using current magnitude minus reference magnitude we get 0.0. However, because the vectors in the two states are at a large angle to each other this result is not very useful. Relative results are only meaningful when displacements in the two states are down a very similar vector.

Modifying a measurement

After you have created a measurement you can modify the selected NODEs (or PARTs) by right-clicking on the textbox and selecting **PICK**

Export to XY_PLOT

Make a graph of distance for the current measurement. Four separate curves will be generated for each measurement containing the X, Y, Z and magnitude components using all time-states in the model. The plots can be saved as curve files or transferred to T/HIS if the T/HIS link is open.

Export ALL to XY_PLOT

Make a graph of distance for all measurement. Four separate curves will be generated for each measurement containing the X, Y, Z and magnitude components using all time-states in the model. The plots can be saved as curve files or transferred to T/HIS if the T/HIS link is open.

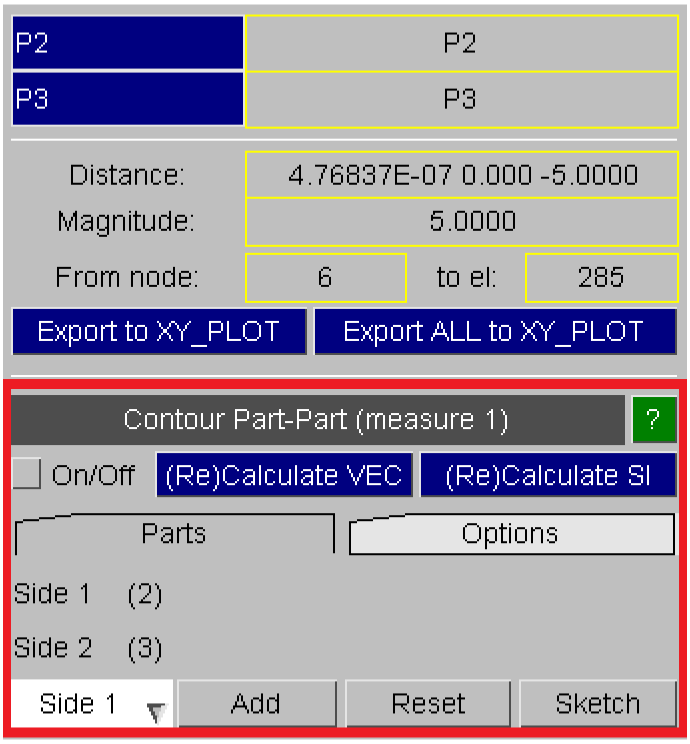
Resuming a Measurement after using Quick Pick

If you press the shortcut key 'q' to swap from measurement picking to quick pick, e.g. to blank some parts the menu will change to display the **RESUME MEASUREMENT** button. Click this when you have finished using quick pick.

9.7.1. Measuring Part to Part with Contouring

Measuring part to part with contouring

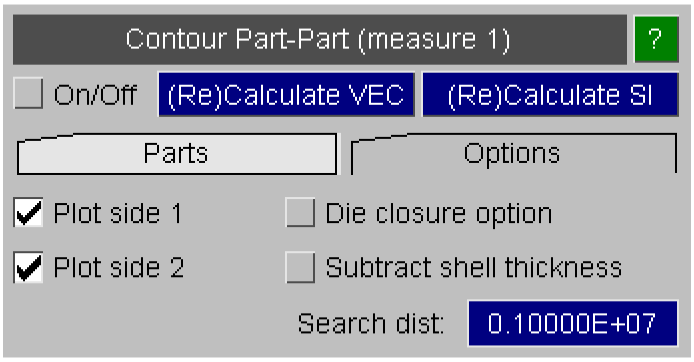
Select two parts to start the normal measure. If it is the first measurement and there is one model in the window, you will be offered the option to contour part-part distance.



For each node, the closest target segment on the other side of the measure will be found, thus defining a vector. This may be plotted as vector (**Vec**) or the scalar magnitude (**SI**).

Users can **Add** more parts (in addition to the two original ones) for either side of the measure.

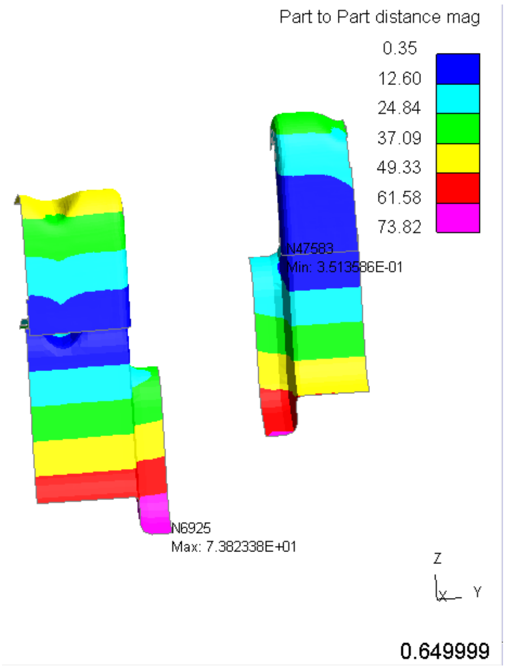
There are further options:



- **Die closure option** will only plot cases where the vector aligns (within 10 degrees) with the node normal (as derived from the surrounding shells)

- **Subtract shell thickness** will remove half the segment thickness at each end if this quantity is known.

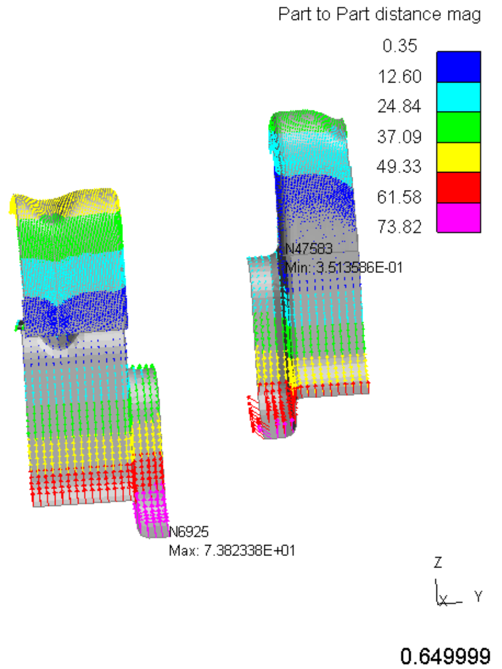
- **Search dist** sets the maximum distance in model space to look for nearby parts. The allows the calculation for very large models to be sped up by reducing the search distance. (Default 1E7)



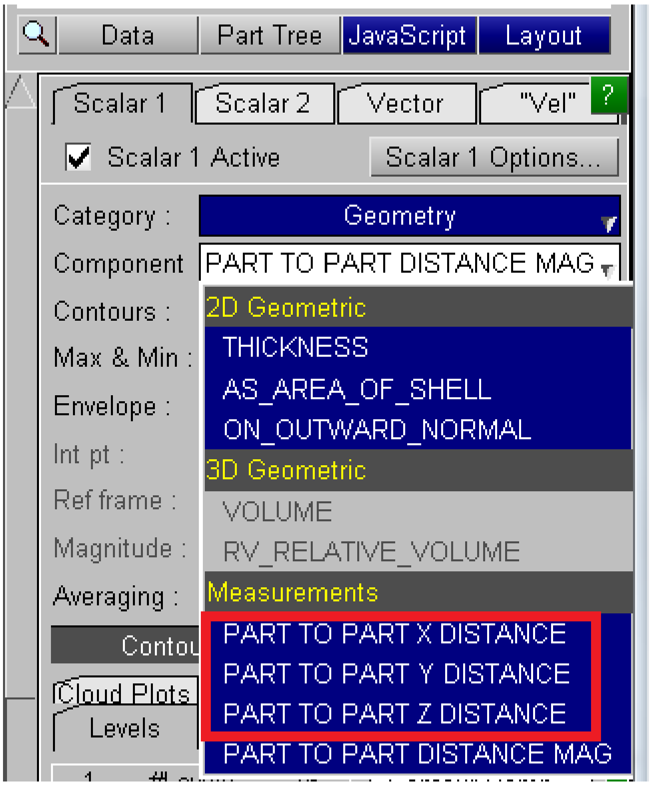
(Re)Calculate SI will give a scalar plot of magnitude of the smallest distance for every node on side 1 vs every segment on side 2 (and vice versa).

Animating the model will now ensure that the measure data is calculated and cached for every state.

The image can then be modified (e.g. by blanking), the correct state selected and the plot recaptured by **CT, SI** or **Vec** .

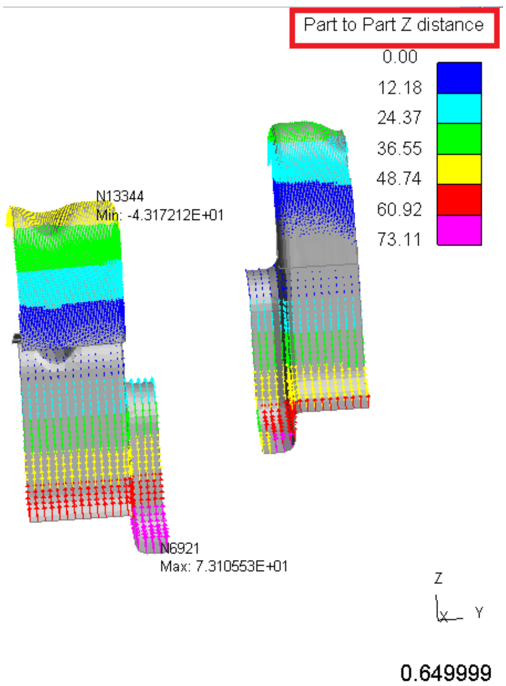


(Re)Calculate VEC will plot the (scaled) vector for each case of closest node vs segment.



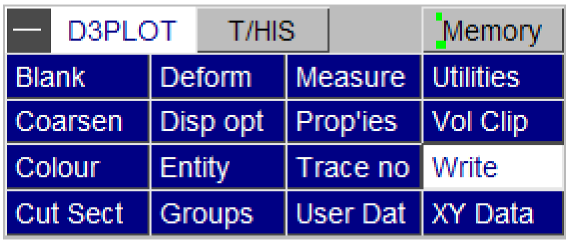
The X, Y, or Z vector component may be plotted instead of the magnitude by accessing the **Geometry** components in the **DATA** menu.

NOTE: This is the X, Y or Z component of the vector defining the shortest node to segment distance which is not the same as the shortest distance in X, etc.



9.8. WRITE Listing Numerical Data to Screen and/or File

WRITE Listing numerical data to screen and/or file



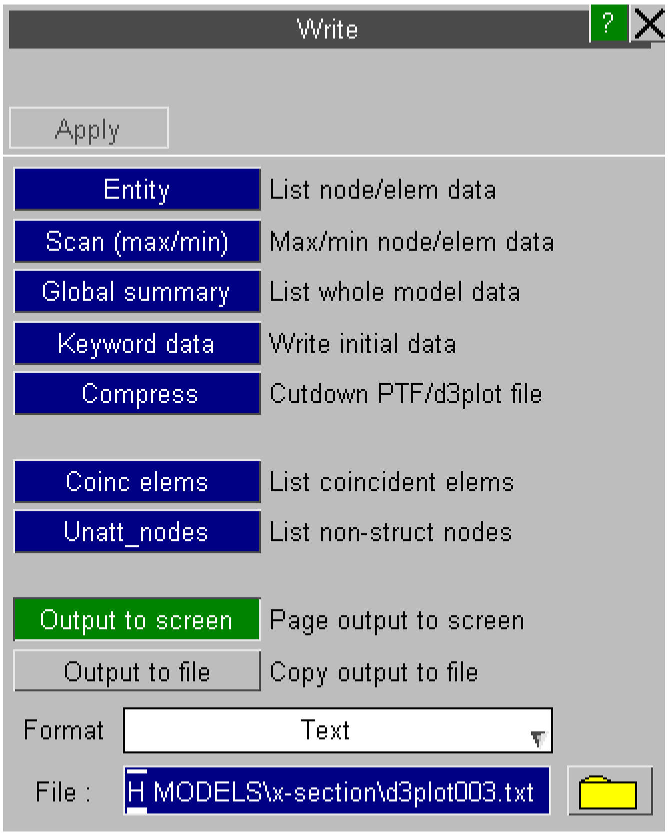
The **WRITE** command allows you to write to the screen and/or file just about any piece of information in the database files accessed by D3PLOT. It is designed to list information at a given time, if you want to plot data (in XY graphical form) you should use the **XY_DATA** command described in XY_DATA Drawing numerical data as XY plots and/or writing it to file instead.

The main **WRITE** control panel is shown right.

The **ENTITY, SCAN, GLOBAL SUMMARY, KEYWORD DATA, COMPRESS, COINC ELEMENTS** and **UNATT_NODES** commands generate output, as described in Sections [WRITE] ENTITY to [WRITE] CUTDOWN D3PLOT/PTF File below.

The **OUTPUT_TO...** options define where the results are to be written. See OUTPUT_TO... Choosing output destination(s) for WRITE output below.

File: is the name of the disk output file.



9.8.1. OUTPUT_TO... Choosing Output Destination(s) for WRITE Output

OUTPUT_TO... Choosing output destination(s) for WRITE output

OUTPUT_TO_SCREEN

Lists the output to a table.

OUTPUT_TO_FILE

Output to a file either in CSV, TEXT or Excel XLSX format. Output of any length may be written.

Only one option can be selected at a time.

9.8.2. [WRITE] ENTITY

[WRITE] ENTITY

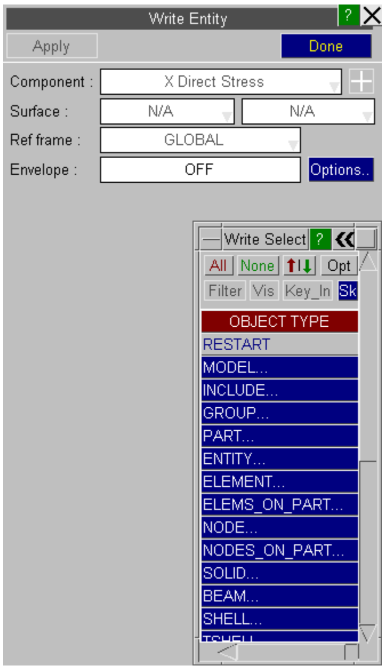
The figure (right) shows the **ENTITY** control panel.

"Entity" output means results for a <list> of standard entity types, in this example 90 solid elements have been chosen, at the current time.

It is also possible to write "global" (ie whole model), "material" and "surface" related information: see Special "entity" types below.

To use this panel:

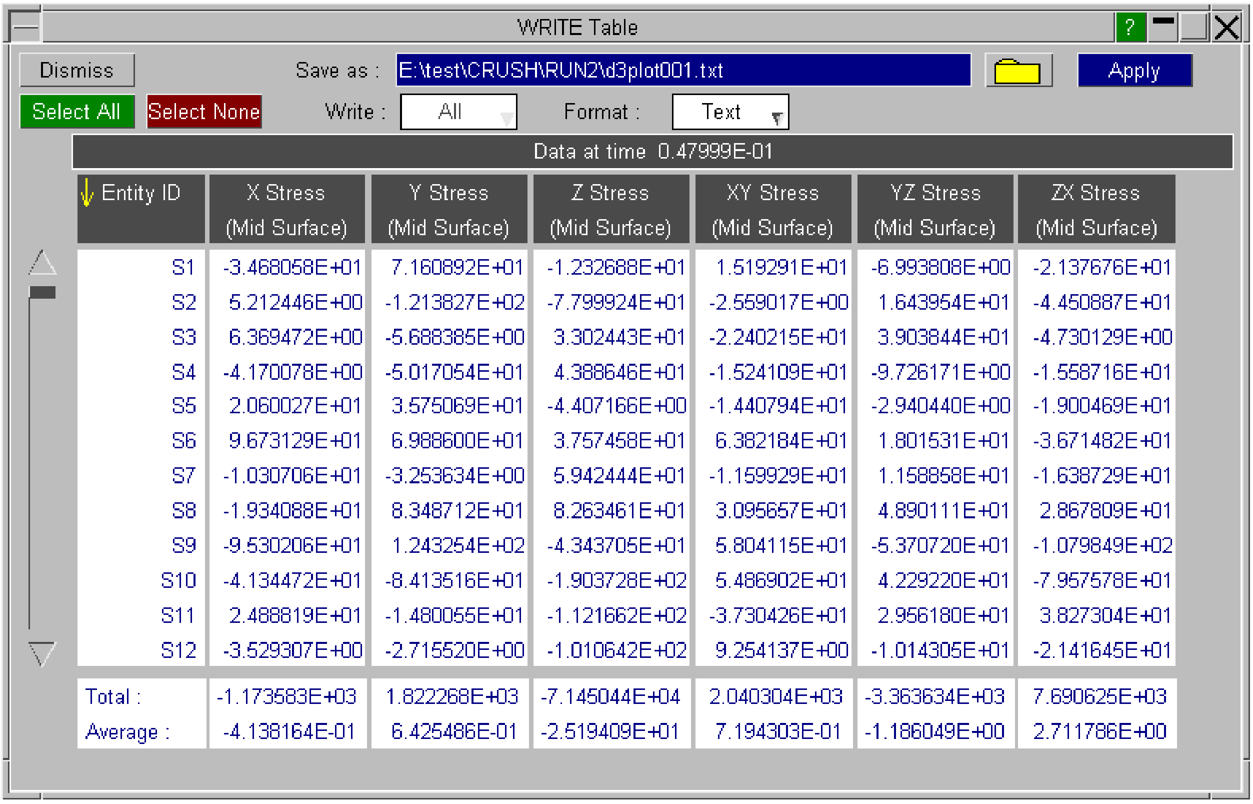
- Select an entity type (eg **NODE**)
- Define a <list> of entities
- Define the data component etc
- Press **APPLY** to generate output



9.8.2.1. What Does "Entity" Output Produce?

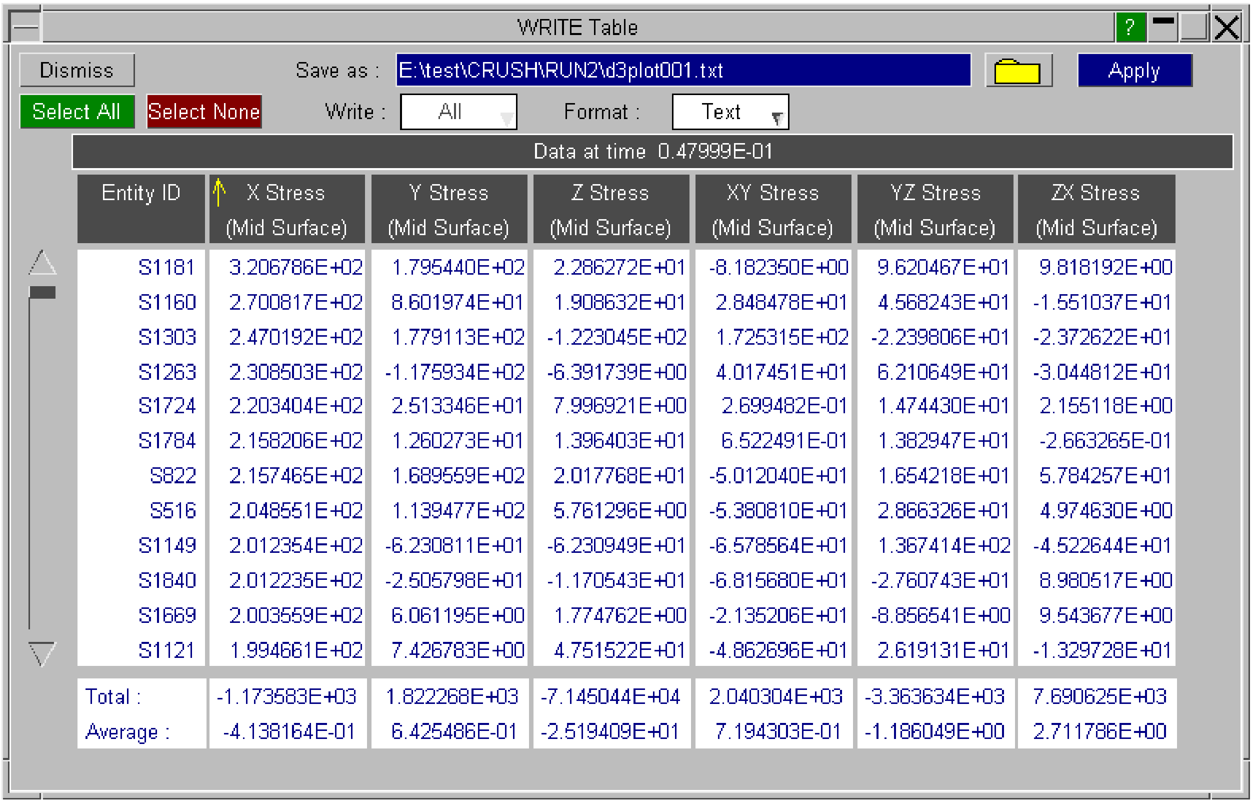
What does "entity" output produce?

You can write out just about any piece of information about any entity in your database, the data is presented in a table. The following example shows the stress tensor for a number of shell elements.



Entity ID	X Stress (Mid Surface)	Y Stress (Mid Surface)	Z Stress (Mid Surface)	XY Stress (Mid Surface)	YZ Stress (Mid Surface)	ZX Stress (Mid Surface)
S1	-3.468058E+01	7.160892E+01	-1.232688E+01	1.519291E+01	-6.993808E+00	-2.137676E+01
S2	5.212446E+00	-1.213827E+02	-7.799924E+01	-2.559017E+00	1.643954E+01	-4.450887E+01
S3	6.369472E+00	-5.688385E+00	3.302443E+01	-2.240215E+01	3.903844E+01	-4.730129E+00
S4	-4.170078E+00	-5.017054E+01	4.388646E+01	-1.524109E+01	-9.726171E+00	-1.558716E+01
S5	2.060027E+01	3.575069E+01	-4.407166E+00	-1.440794E+01	-2.940440E+00	-1.900469E+01
S6	9.673129E+01	6.988600E+01	3.757458E+01	6.382184E+01	1.801531E+01	-3.671482E+01
S7	-1.030706E+01	-3.253634E+00	5.942444E+01	-1.159929E+01	1.158858E+01	-1.638729E+01
S8	-1.934088E+01	8.348712E+01	8.263461E+01	3.095657E+01	4.890111E+01	2.867809E+01
S9	-9.530206E+01	1.243254E+02	-4.343705E+01	5.804115E+01	-5.370720E+01	-1.079849E+02
S10	-4.134472E+01	-8.413516E+01	-1.903728E+02	5.486902E+01	4.229220E+01	-7.957578E+01
S11	2.488819E+01	-1.480055E+01	-1.121662E+02	-3.730426E+01	2.956180E+01	3.827304E+01
S12	-3.529307E+00	-2.715520E+00	-1.010642E+02	9.254137E+00	-1.014305E+01	-2.141645E+01
Total :	-1.173583E+03	1.822268E+03	-7.145044E+04	2.040304E+03	-3.363634E+03	7.690625E+03
Average :	-4.138164E-01	6.425486E-01	-2.519409E+01	7.194303E-01	-1.186049E+00	2.711786E+00

By default the data is displayed in order of increasing entity ID but the data can also be sorted by any column into either increasing or decreasing order by clicking on the column header.



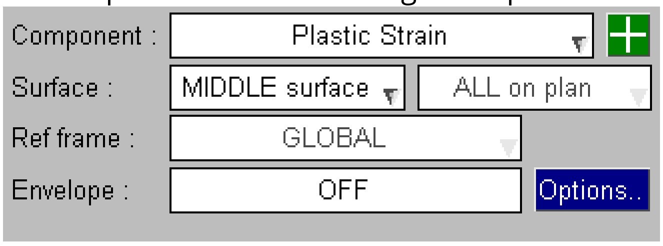
Entity ID	X Stress (Mid Surface)	Y Stress (Mid Surface)	Z Stress (Mid Surface)	XY Stress (Mid Surface)	YZ Stress (Mid Surface)	ZX Stress (Mid Surface)
S1181	3.206786E+02	1.795440E+02	2.286272E+01	-8.182350E+00	9.620467E+01	9.818192E+00
S1160	2.700817E+02	8.601974E+01	1.908632E+01	2.848478E+01	4.568243E+01	-1.551037E+01
S1303	2.470192E+02	1.779113E+02	-1.223045E+02	1.725315E+02	-2.239806E+01	-2.372622E+01
S1263	2.308503E+02	-1.175934E+02	-6.391739E+00	4.017451E+01	6.210649E+01	-3.044812E+01
S1724	2.203404E+02	2.513346E+01	7.996921E+00	2.699482E-01	1.474430E+01	2.155118E+00
S1784	2.158206E+02	1.260273E+01	1.396403E+01	6.522491E-01	1.382947E+01	-2.663265E-01
S822	2.157465E+02	1.689559E+02	2.017768E+01	-5.012040E+01	1.654218E+01	5.784257E+01
S516	2.048551E+02	1.139477E+02	5.761296E+00	-5.380810E+01	2.866326E+01	4.974630E+00
S1149	2.012354E+02	-6.230811E+01	-6.230949E+01	-6.578564E+01	1.367414E+02	-4.522644E+01
S1840	2.012235E+02	-2.505798E+01	-1.170543E+01	-6.815680E+01	-2.760743E+01	8.980517E+00
S1669	2.003559E+02	6.061195E+00	1.774762E+00	-2.135206E+01	-8.856541E+00	9.543677E+00
S1121	1.994661E+02	7.426783E+00	4.751522E+01	-4.862696E+01	2.619131E+01	-1.329728E+01
Total :	-1.173583E+03	1.822268E+03	-7.145044E+04	2.040304E+03	-3.363634E+03	7.690625E+03
Average :	-4.138164E-01	6.425486E-01	-2.519409E+01	7.194303E-01	-1.186049E+00	2.711786E+00

9.8.2.2. Selecting an Entity Type

Selecting an entity type

Firstly choose the type of entity for which you want to extract data. In this example the user has chosen **Shells**.

After selecting the entity type the options to select the data component, shell surface and frame of reference will become active. If the model contains results at on plan integration points for fully integrated shells or at all integration points for fully integrated solids then the option to select an integration point will also become active.

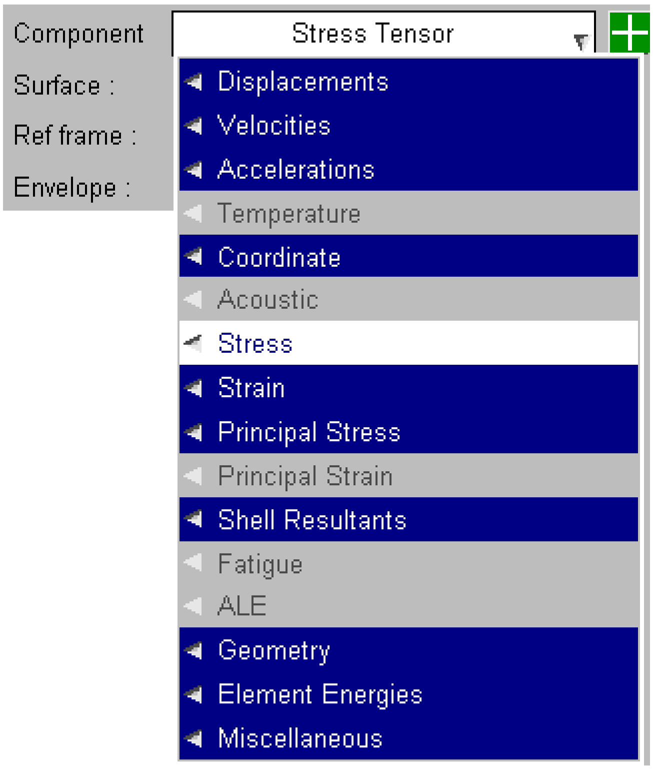


9.8.2.3. COMPONENT Choosing a Data Component

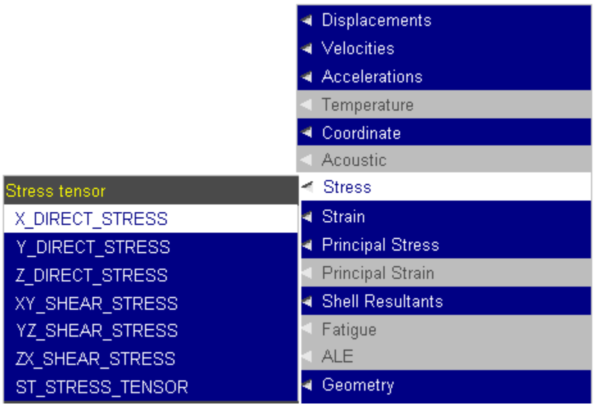
COMPONENT Choosing a data component

Every entity type has at least one data component associated with it, and most have several.

To select the data component use the popup menu to select first the data component category.



and then the data component



9.8.2.4. SURFACE Shell Surface Selection

SURFACE Shell surface selection

If Shell elements are selected, the **SURFACE** button is used to determine for which surface the results will be written.

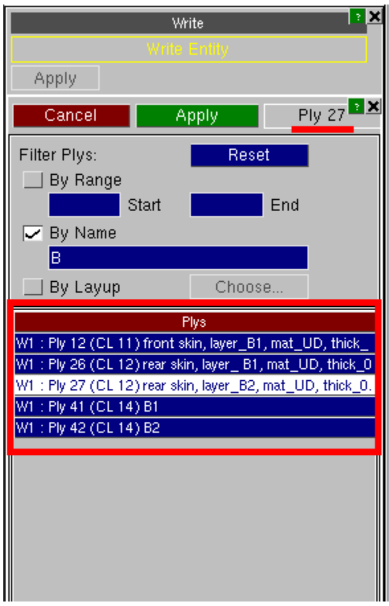
You are referred to SURFACE / INT Point for a basic description of shell surfaces and Thin shell integration points for a more detailed description.

Note 1: You have to define surfaces for thick shell output in **WRITE** . This is not necessary when plotting since it is possible to draw all three sets of values simultaneously.

Note 2: Top and bottom shell surfaces are not the element "outer fibres" if the default Gaussian integration scheme is used. See Thin shell integration points for more details.

Note 3: It is possible to make LS-DYNA generate output at other than the three surfaces described here, in which case "layers" will be available for selection as well as surfaces. Typically this will be used for a composites analyses: see Thin shell integration points.

SURFACE with composite plys



From D3PLOT 13.0 onwards, if a model contains composite plys then results can be written for the surface which corresponds to a ply (requires a `.ztf` file). Under the **SURFACE** button choose **Select plys...** . A single ply can be selected from the menu.

The currently selected ply is shown in the top right. The user can select a different ply from the list of available plys.

The list of available plys can be filtered:

By Range	Ply IDs must lie within the given range.
By Name	Ply names must contain this text (case insensitive).
By Layup	Plys must be contained in selected layups. (Only available if layups have been set-up in PRIMER.)
Reset	Deselects **By Range** , **By Name** and **By Layup,** and clears start, end and name fields, and any selected layups.

The plys are ordered by layup ID and position in layup (where layups have been set-up), or by ply ID (if layups are not available).

9.8.2.5. FRAME_OF_REF Local, Global or Cylindrical

FRAME_OF_REF Local, global or cylindrical

When a directional stress or strain component is chosen for solids, shells or thick shells the frame of reference for output must be defined.

This defaults to global, but you can opt for element local, (global) cylindrical, or user-defined coordinate systems. If the model has composite plys there is also a ply local option.

You are referred to REF_FRAME... Choosing the Frame of Reference for a basic description of frame of reference.

Note:

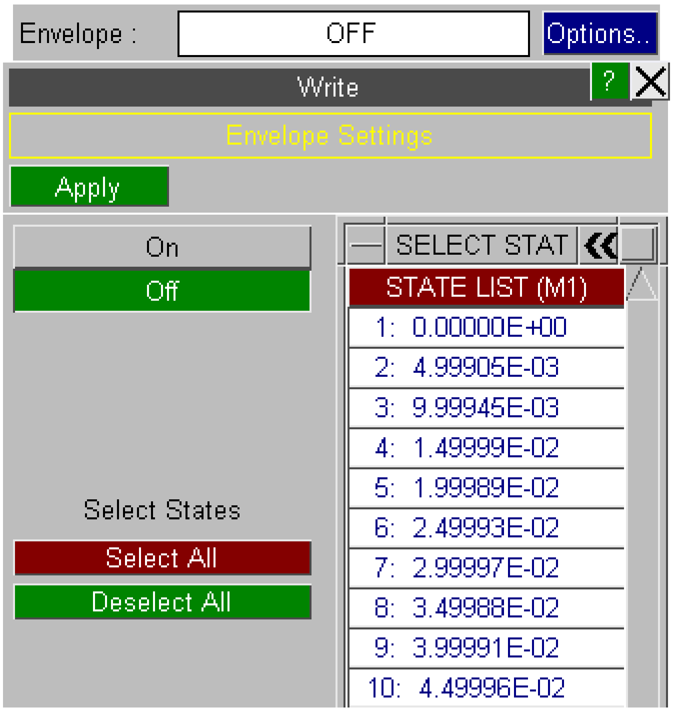
It is possible, but unusual, to make LS-DYNA write stresses for some material models in the material axis system. This would usually only be used for the output of orthotropic material results and laminates. See Cases Where Stress and Strain Tensor Output are in the Local Coordinate System for more details.

There is no way to tell from the database file that directional stresses for these element types have been written this way, and D3PLOT assumes that they are in the global system: **it is your responsibility to interpret your results correctly.**

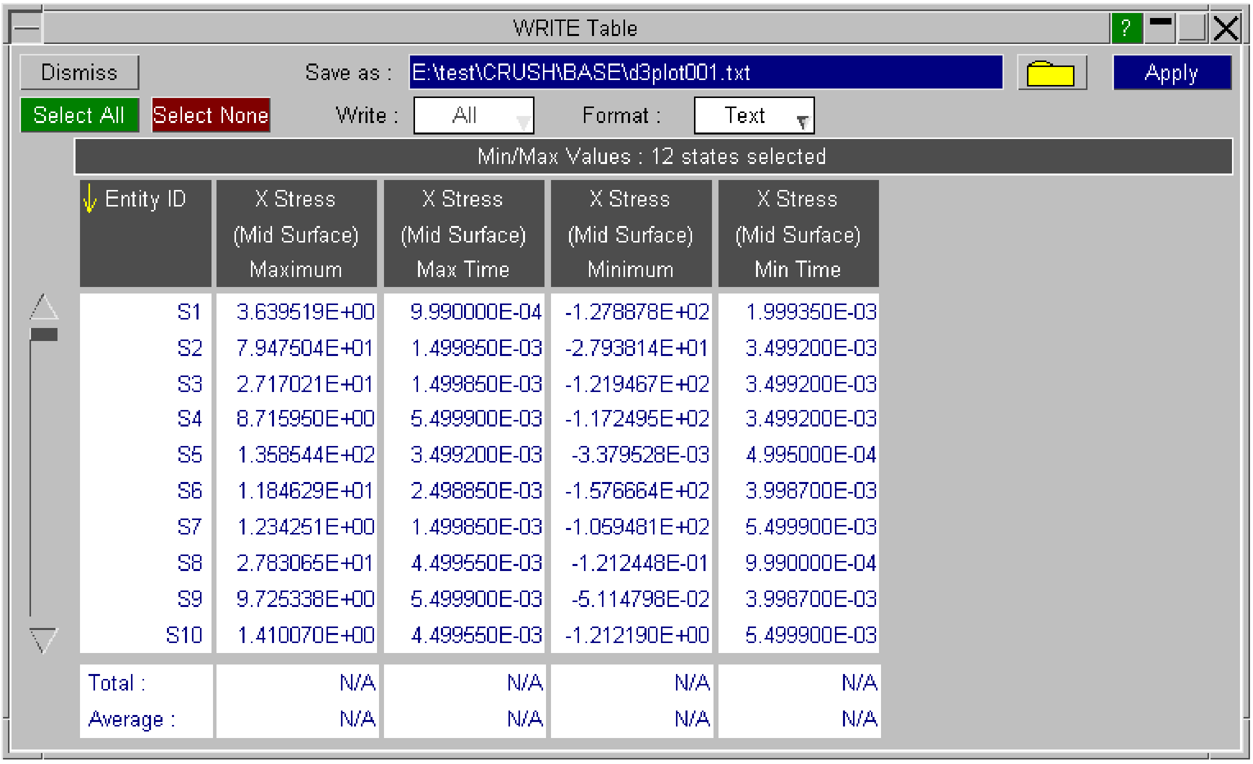
9.8.2.6. ENVELOPE

ENVELOPE

Instead of reporting results at a single time-state, D3PLOT can calculate the maximum or minimum values occurring over a selection of time-states.



Once a component has been selected the user can press **APPLY** and the output will be generated showing both the maximum and minimum values for each entity.



Entity ID	X Stress (Mid Surface) Maximum	X Stress (Mid Surface) Max Time	X Stress (Mid Surface) Minimum	X Stress (Mid Surface) Min Time
S1	3.639519E+00	9.990000E-04	-1.278878E+02	1.999350E-03
S2	7.947504E+01	1.499850E-03	-2.793814E+01	3.499200E-03
S3	2.717021E+01	1.499850E-03	-1.219467E+02	3.499200E-03
S4	8.715950E+00	5.499900E-03	-1.172495E+02	3.499200E-03
S5	1.358544E+02	3.499200E-03	-3.379528E-03	4.995000E-04
S6	1.184629E+01	2.498850E-03	-1.576664E+02	3.998700E-03
S7	1.234251E+00	1.499850E-03	-1.059481E+02	5.499900E-03
S8	2.783065E+01	4.499550E-03	-1.212448E-01	9.990000E-04
S9	9.725338E+00	5.499900E-03	-5.114798E-02	3.998700E-03
S10	1.410070E+00	4.499550E-03	-1.212190E+00	5.499900E-03
Total :	N/A	N/A	N/A	N/A
Average :	N/A	N/A	N/A	N/A

9.8.2.7. Averaged and Unaveraged Element Results Written at Nodes

Averaged and Unaveraged element results written at nodes

Element data written at nodes has to be averaged. All elements attached to that node which could legitimately contribute the required data component are included, and the unweighted average is used. If elements are blanked or volume-clipped they are only excluded if the **MATERIAL_IGNORED** or **CLIPPING_IGNORED s** witches respectively are on.

Problems can arise in the following situations:

On material boundaries: For the node there is no "parent" element to give a definitive material number, as there is when plotting, so averaging will always take place across dissimilar materials. At element type junctions: For the node there is no "parent" element type. So where, for example, a node is common to a solid and a shell, the results will be averaged across the two types.

Therefore you should take care when using **WRITE** to extract element data averaged at nodes.

Whenever **WRITE** tabulates data it prefaces it with its full data component name and, for nodal results, it also states whether the figures are "averaged" or "unaveraged".

9.8.2.8. Special "Entity" Types

Special "entity" types

As well as the standard node and element categories, familiar in other contexts, you can select the following "global" and other information for output:

GLOBAL

Refers to the whole model. LS-DYNA computes energies, rigid body velocities and mass; all of which may be output. See Global (Whole Model) Data Components for more details.

PART Refers to solid, shell, beam and thick-shell (but not spring) **PART** s. LS-DYNA calculates energies, velocities and mass for these; all of which may be output. See Part ("Material") Data Components for more details. SURFACE

Refers to contact surfaces. LS-DYNA writes nodal forces to the `.CTF` file, and D3PLOT can sum these to produce overall forces on a surface. (You can also choose to compute **SURFACE A** or **B** side forces only.) See Contact Surface Summary Components for more details.

AIRBAG Refers to "airbags" - effectively control volumes - used with the Airbag Particle method. This capability is still under development in LS-DYNA, and only a limited amount of information is available for post-processing, see Airbag Particle (ABP) Data Components for more information. GROUPS Refers to the contents of groups. Because groups can contain a mixture of entities D3PLOT will present a list of components that are valid for the selected group. If a component is selected that is not valid for all the entities in the group, only the valid entities will be listed, e.g. if the component AXIAL_FORCE is selected for a group containing both beams and shells, only the beams will be listed. See GROUPS for more information. INCLUDES

Refers to the contents of include files. Include files are similar to groups in that they can contain a mixture of entities and the same rules apply.

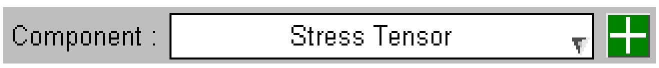
Output for these categories of information under ENTITY is selective: you can choose those materials or surfaces for which you want to see results.

To list complete summary information you can use the GLOBAL_SUMMARY option in the main WRITE menu instead: this summarises data for all materials or all contact surfaces, which may be more convenient.

9.8.2.9. Selecting Multiple Components

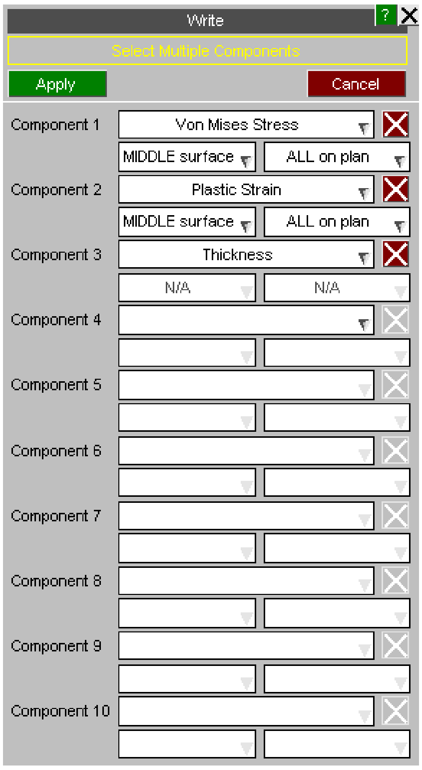
Selecting Multiple Components

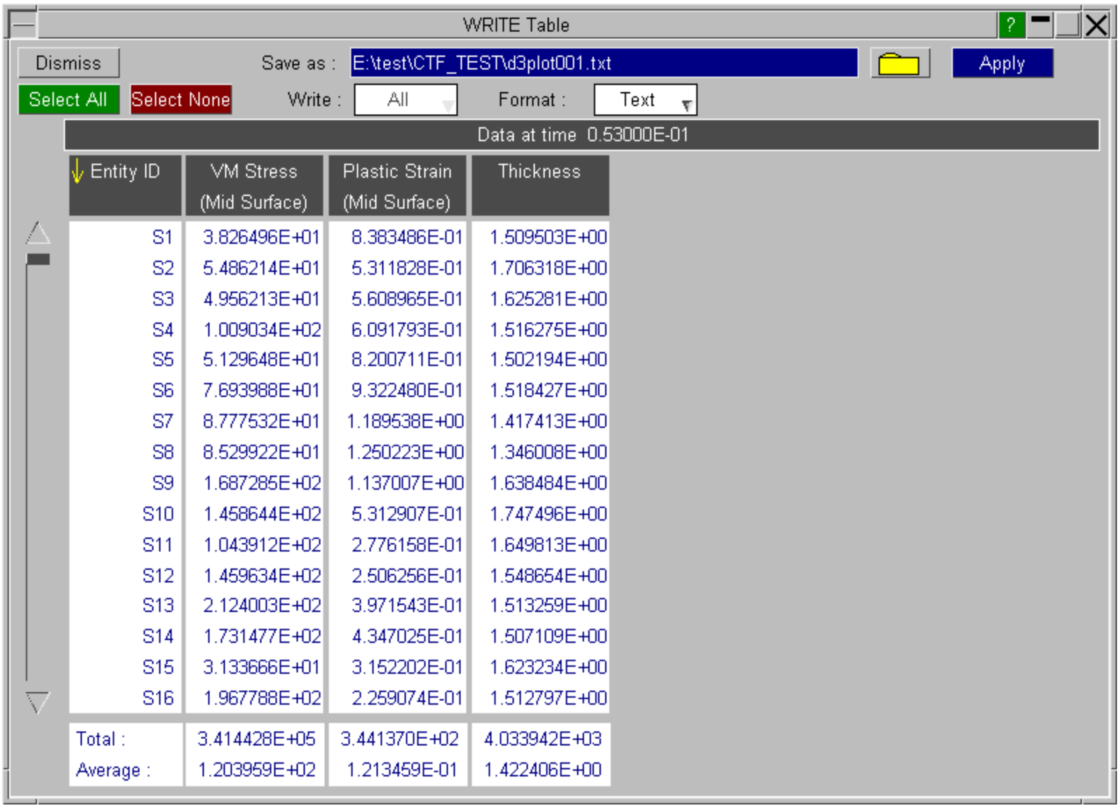
From D3PLOT 13.0 onwards D3PLOT the WRITE menu can display multiple data components. To select multiple components select the next to the data component popup.



Up to 10 different data components can be selected and displayed at the same time.

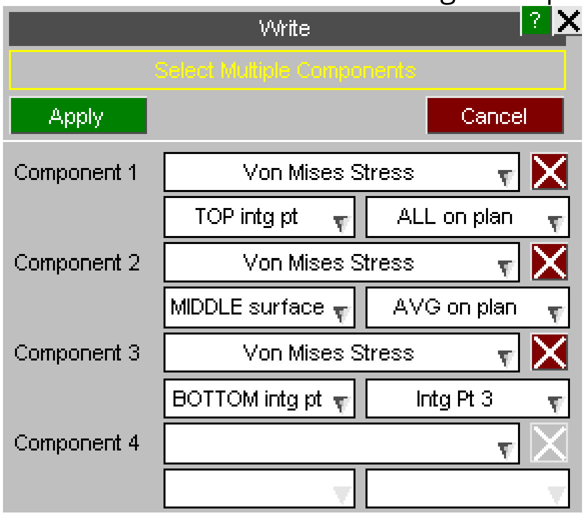
The can be used to remove a data component from the list.

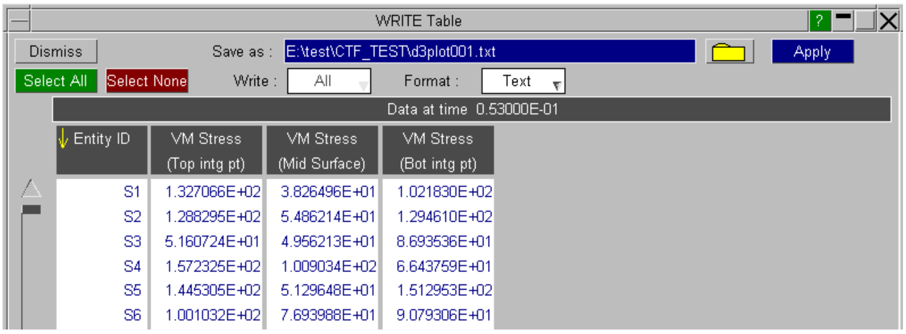




Entity ID	VM Stress (Mid Surface)	Plastic Strain (Mid Surface)	Thickness
S1	3.826496E+01	8.383486E-01	1.509503E+00
S2	5.486214E+01	5.311828E-01	1.706318E+00
S3	4.956213E+01	5.608965E-01	1.625281E+00
S4	1.009034E+02	6.091793E-01	1.516275E+00
S5	5.129648E+01	8.200711E-01	1.502194E+00
S6	7.693988E+01	9.322480E-01	1.518427E+00
S7	8.777532E+01	1.189538E+00	1.417413E+00
S8	8.529922E+01	1.250223E+00	1.346008E+00
S9	1.687285E+02	1.137007E+00	1.638484E+00
S10	1.458644E+02	5.312907E-01	1.747496E+00
S11	1.043912E+02	2.776158E-01	1.649813E+00
S12	1.459634E+02	2.506256E-01	1.548654E+00
S13	2.124003E+02	3.971543E-01	1.513259E+00
S14	1.731477E+02	4.347025E-01	1.507109E+00
S15	3.133666E+01	3.152202E-01	1.623234E+00
S16	1.967788E+02	2.259074E-01	1.512797E+00
Total :	3.414428E+05	3.441370E+02	4.033942E+03
Average :	1.203959E+02	1.213459E-01	1.422406E+00

As well as being able to select different data components the same component can be selected multiple times but with different surface / integration points.



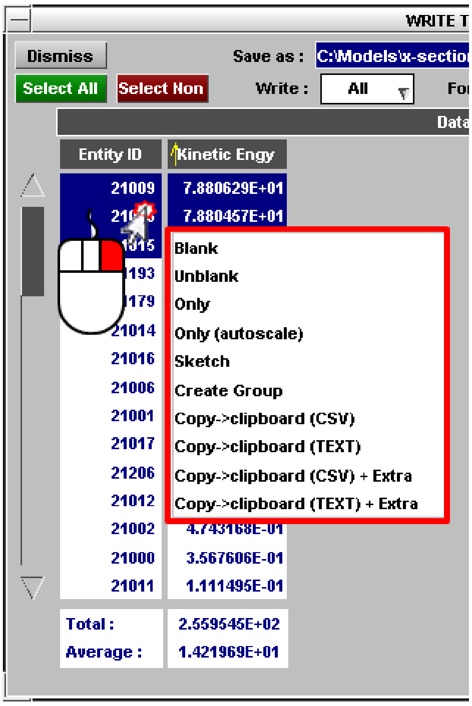


9.8.2.10. Write Table Options

Write Table Options

There are a number of options in the table that allow you to output the data to file and display the entities in the D3PLOT graphics window.

Each row in the table can be selected by left-clicking on it and the shift and ctrl keys can be used to select multiple rows. By right-clicking on the selection you can carry out the following actions:



The first set of options can be used to visualise the selected items:

BLANK	Blank the selected items
UNBLANK	Unblank the selected items
ONLY	Only the selected items
ONLY (Autoscale)	Only the selected items and Autoscale them
SKETCH	Sketch the selected items
CREATE GROUP	Creates a group which can then be written to a file, modified or sketched in the Groups menu

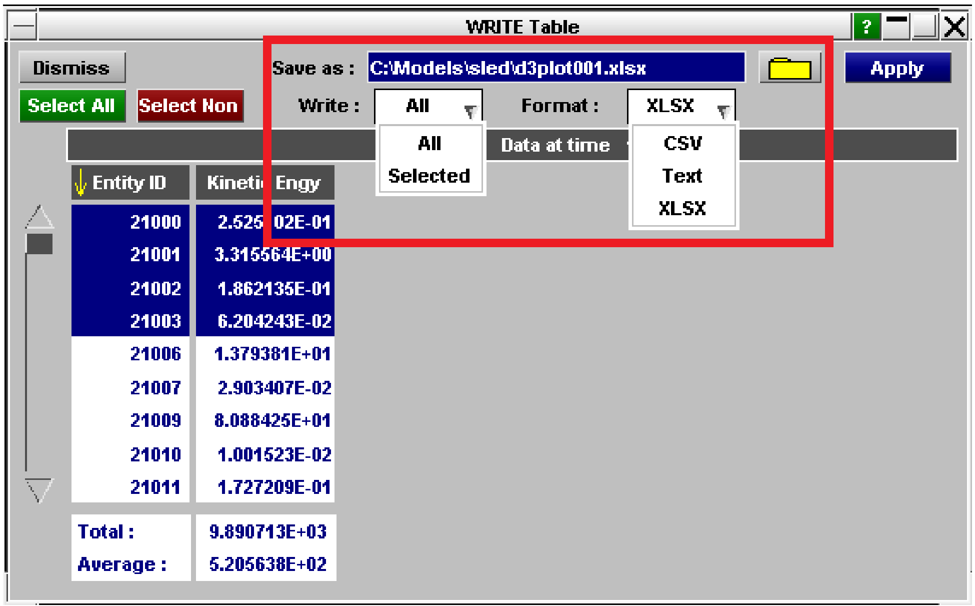
The second set of options can be used to copy and paste the selected text to other programs:

COPY->CLIPBOARD (CSV)	Copy the selected items to the clipboard in CSV format.
COPY->CLIPBOARD (TEXT)	Copy the selected items to the clipboard in TEXT format (columns separated by spaces).

COPY->CLIPBOARD (CSV) + EXTRA Copy the selected items to the clipboard in CSV format. Also copies the headers, total and average.

COPY->CLIPBOARD (TEXT) + EXTRA Copy the selected items to the clipboard in TEXT format (columns separated by spaces). Also copies the headers, total and average.

The data displayed in the table can be written to file in CSV, TEXT or Excel XLSX format. You can select to write all the data or just the selected items.



9.8.3. [WRITE] SCAN

[WRITE] SCAN

This figure (right) shows the **SCAN c** ontrol panel.

"Scan" output means the maximum and minimum results for a given entity/data component combination.

In this example the user is **SCAN** ning for the max/min Global X stress in solids.

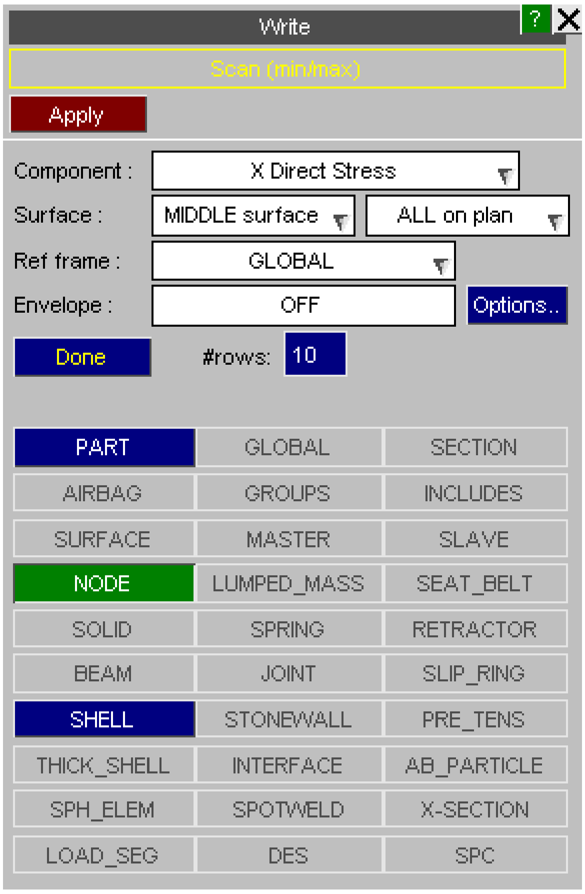
To use this panel:

- Select an entity type (eg **NODE**)

- Define the data component etc

- Define **#rows** to calculate

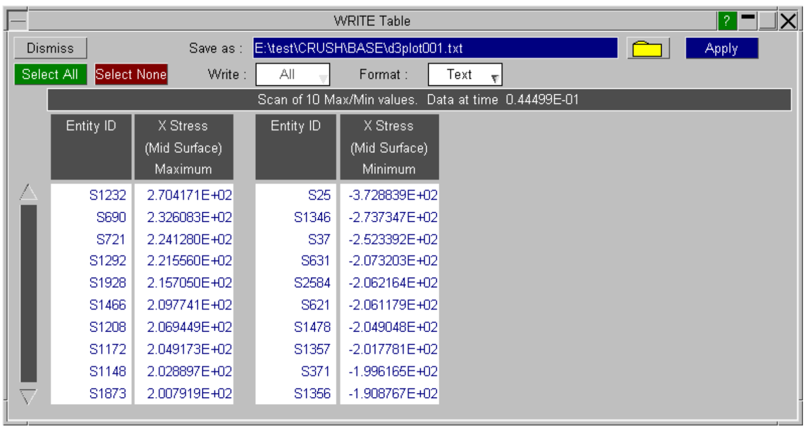
- Press **APPLY** to generate output



9.8.3.1. What Does a "Scan" Produce?

What does a "scan" produce?

A "scan" operation searches the list of entities defined, and outputs the <#rows> maximum and minimum scalar values that are eligible for plotting (see What Does "Eligible For Plotting" Mean? below) in a table. An example of typical output is shown below:



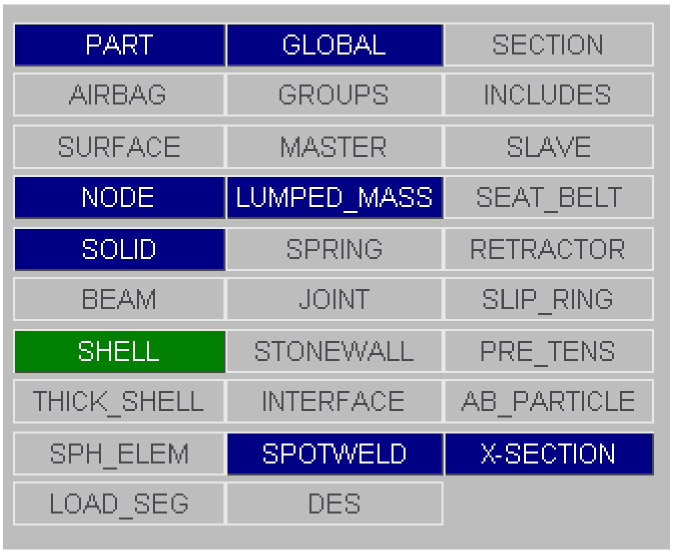
Entity ID	X Stress (Mid Surface) Maximum	Entity ID	X Stress (Mid Surface) Minimum
S1232	2.704171E+02	S25	-3.728839E+02
S690	2.326083E+02	S1346	-2.737347E+02
S721	2.241280E+02	S37	-2.523392E+02
S1292	2.215560E+02	S631	-2.073203E+02
S1928	2.157050E+02	S2584	-2.062164E+02
S1466	2.097741E+02	S621	-2.061179E+02
S1208	2.069449E+02	S1478	-2.049048E+02
S1172	2.049173E+02	S1357	-2.017781E+02
S1148	2.028897E+02	S371	-1.996165E+02
S1873	2.007919E+02	S1356	-1.908767E+02

9.8.3.2. Selecting an Entity Type

Selecting an entity type

Firstly choose the type of entity for which you want to extract data. In this example the user has chosen **Shells** .

In SCAN mode all entities of the chosen type which are eligible for plotting (see What does "eligible for plotting" mean? below) are included in the scan operation.



9.8.3.3. COMPONENT Selecting a Data Component

COMPONENT Selecting a data component

This is done in exactly the same way as for **[WRITE] ENTITY** output, see COMPONENT Choosing a data component , but only scalar data components are permitted.

9.8.3.4. SURFACE Selecting a Shell Surface

SURFACE Selecting a shell surface

This is done in exactly the same way as for **[WRITE] ENTITY** output, see SURFACE Shell surface selection .

9.8.3.5. FRAME_OF_REF Selecting Global/Local/Cylindrical Frame Of Reference

FRAME_OF_REF Selecting global/local/cylindrical frame of reference

This is done in exactly the same way as for **[WRITE] ENTITY** output, see FRAME_OF_REF Local, global or cylindrical .

9.8.3.6. What Does "Eligible For Plotting" Mean?

What does "eligible for plotting" mean?

Entities are only included in a "scan" operation if they are currently available for plotting. This permits you to restrict the range of entities scanned using the same tools that are used to restrict what is plotted. The things that exclude them are: Blanking:

Blanked nodes/elements are excluded. The contribution of blanked elements to averaging data at nodes depends on the **BLANKING_IGNORED** switch.

Vol clipping:

Volume-clipped nodes/elements are excluded. The contribution of clipped elements to averaging data at nodes depends on the **CLIPPING_IGNORED** switch.

Note that an element which is off the current screen because of scale or view-point is still "eligible" for plotting.

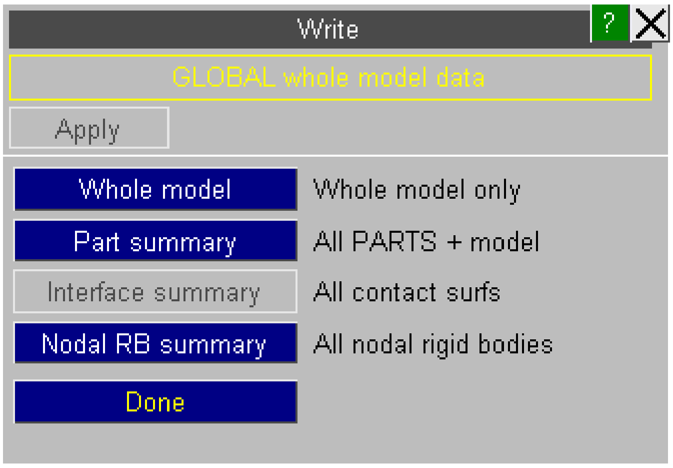
9.8.4. [WRITE] GLOBAL_SUMMARY

[WRITE] GLOBAL_SUMMARY

This figure shows the Global summary control panel.

This controls the display of the summaries of "global" (whole model, material and contact surface) data.

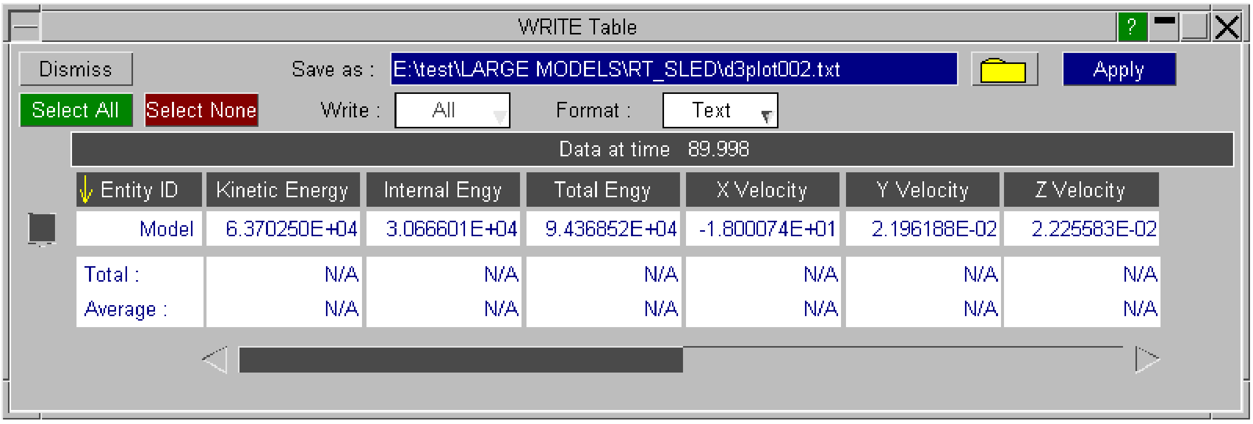
This data can also be extracted in a more selective fashion using the **ENTITY** command.



9.8.4.1. WHOLE_MODEL Summary Data About the Whole Model Only

WHOLE_MODEL Summary data about the whole model only

This is a summary of the energies and rigid-body velocities of the whole model:

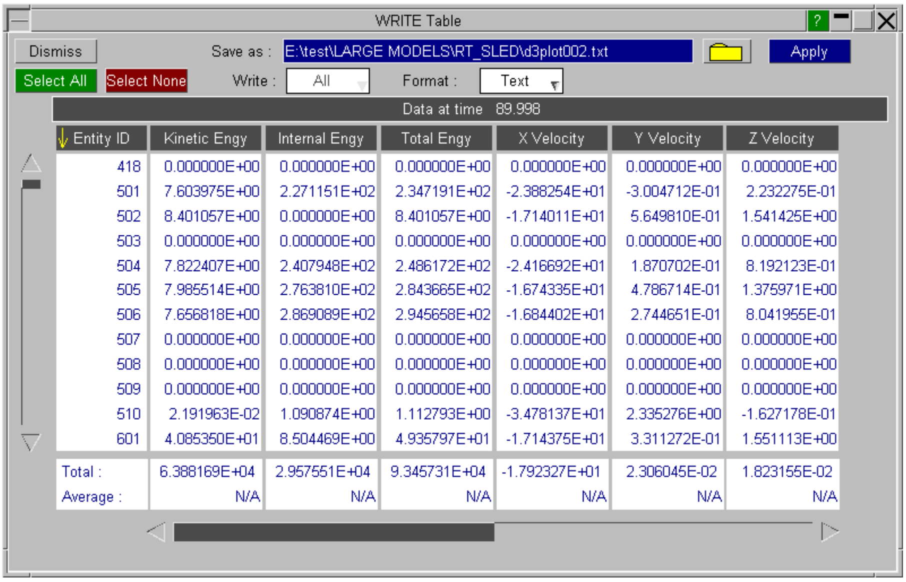


Entity ID	Kinetic Energy	Internal Engy	Total Engy	X Velocity	Y Velocity	Z Velocity
Model	6.370250E+04	3.066601E+04	9.436852E+04	-1.800074E+01	2.196188E-02	2.225583E-02
Total :	N/A	N/A	N/A	N/A	N/A	N/A
Average :	N/A	N/A	N/A	N/A	N/A	N/A

9.8.4.2. PART_SUMMARY Summary Data for All PARTS + Whole Model

PART_SUMMARY Summary data for all PARTS + whole model

This is the same as above, but broken down by material.



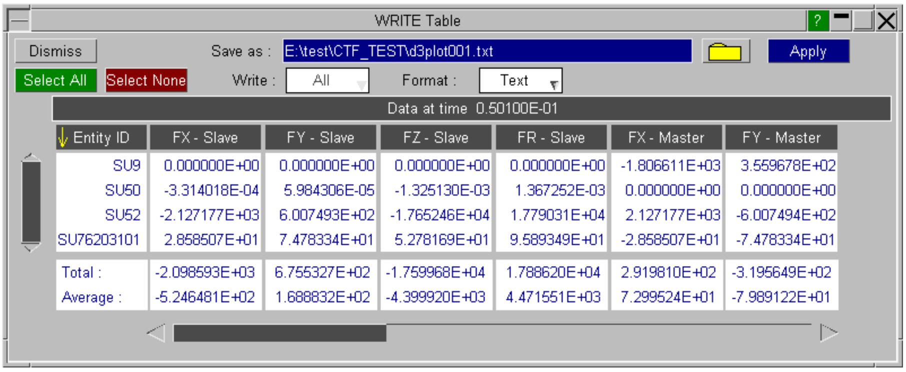
Entity ID	Kinetic Engy	Internal Engy	Total Engy	X Velocity	Y Velocity	Z Velocity
418	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
501	7.603975E+00	2.271151E+02	2.347191E+02	-2.388254E+01	-3.004712E-01	2.232275E-01
502	8.401057E+00	0.000000E+00	8.401057E+00	-1.714011E+01	5.649810E-01	1.541425E+00
503	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
504	7.822407E+00	2.407948E+02	2.486172E+02	-2.416692E+01	1.870702E-01	8.192123E-01
505	7.985514E+00	2.763810E+02	2.843665E+02	-1.674335E+01	4.786714E-01	1.375971E+00
506	7.656818E+00	2.869089E+02	2.945658E+02	-1.684402E+01	2.744651E-01	8.041955E-01
507	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
508	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
509	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
510	2.191963E-02	1.090874E+00	1.112793E+00	-3.478137E+01	2.335276E+00	-1.627178E-01
601	4.085350E+01	8.504469E+00	4.935797E+01	-1.714375E+01	3.311272E-01	1.551113E+00
Total :	6.388169E+04	2.957551E+04	9.345731E+04	-1.792327E+01	2.306045E-02	1.823155E-02
Average :	N/A	N/A	N/A	N/A	N/A	N/A

Note: The "Sum" of all material values is frequently less than the (whole) "Model" values above. This is because the "Model" values include the contribution from any lumped-masses, contact friction, etc that are not part of "material" data.

9.8.4.3. INTERFACE_SUMMARY Summary Forces on All Contact Surfaces

INTERFACE_SUMMARY Summary forces on all contact surfaces

This lists the global [x,y,z,magnitude] force vectors for all contact surfaces. The values are computed from the individual nodal forces in the `.CTF` file, and are presented separately for A and B sides of each surface.



Note 1: Normal contact surfaces with both A and B sides will have equal and opposite forces on each side. For example surface #1 here.

Note 2: The contact force file only contains forces for nodes on segments. Where discrete nodes form one side of a surface then no forces are written for that side. For example surface #4 above is "Discrete nodes (A side) impacting surface (B side)", and forces only appear for the B side.

Note 3: Single-surface contacts only have an A side, and the net force on such a contact surface is zero. However plots will show a distribution of force on the individual segments of the contact.

Note 4: Contact surface types using a one-way treatment (eg types 10 and 21) write forces for the B side only. Surface #3 above is such a surface.

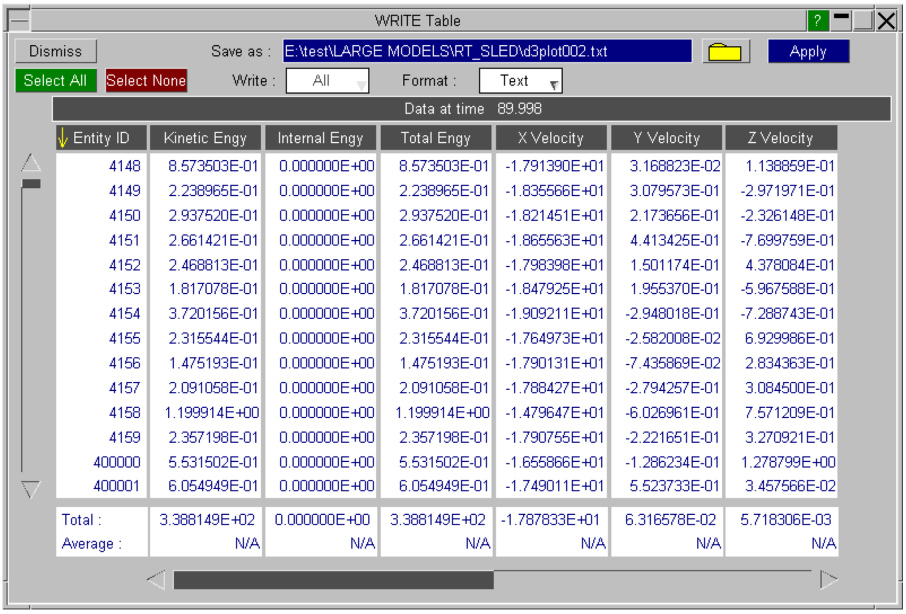
Note 5:

Contacts using a constraint rather than a penalty formulation do not generate forces. (However contacts using "soft constraint" still generate forces.)

9.8.4.4. NODAL_RB_SUMMARY Summary Energy and Velocity for Nodal Rigid Bodies

NODAL_RB_SUMMARY Summary energy and velocity for nodal rigid bodies

Nodal rigid bodies write energy and rigid body results in the same way as materials:



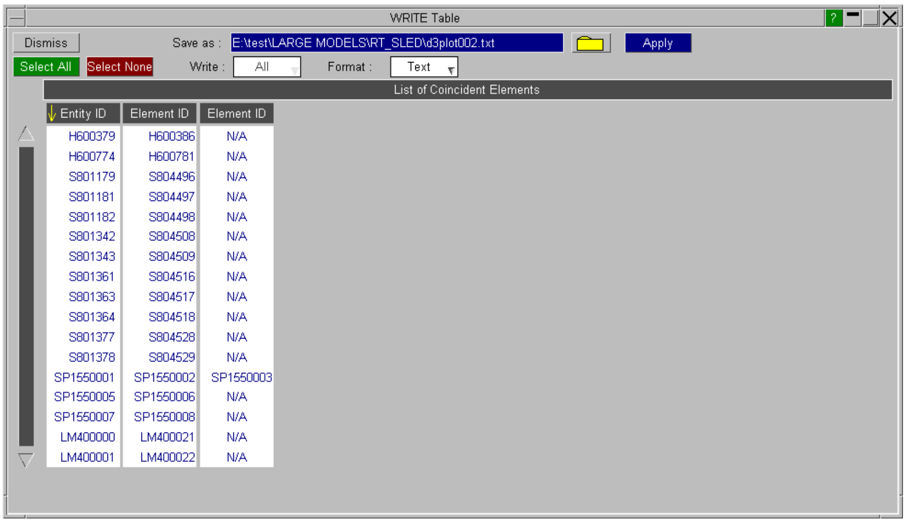
Entity ID	Kinetic Engy	Internal Engy	Total Engy	X Velocity	Y Velocity	Z Velocity
4148	8.573503E-01	0.000000E+00	8.573503E-01	-1.791390E+01	3.168823E-02	1.138859E-01
4149	2.238965E-01	0.000000E+00	2.238965E-01	-1.835566E+01	3.079573E-01	-2.971971E-01
4150	2.937520E-01	0.000000E+00	2.937520E-01	-1.821451E+01	2.173656E-01	-2.326148E-01
4151	2.661421E-01	0.000000E+00	2.661421E-01	-1.865563E+01	4.413425E-01	-7.699759E-01
4152	2.468813E-01	0.000000E+00	2.468813E-01	-1.798398E+01	1.501174E-01	4.378084E-01
4153	1.817078E-01	0.000000E+00	1.817078E-01	-1.847925E+01	1.955370E-01	-5.967588E-01
4154	3.720156E-01	0.000000E+00	3.720156E-01	-1.909211E+01	-2.948018E-01	-7.288743E-01
4155	2.315544E-01	0.000000E+00	2.315544E-01	-1.764973E+01	-2.582008E-02	6.929986E-01
4156	1.475193E-01	0.000000E+00	1.475193E-01	-1.790131E+01	-7.435869E-02	2.834363E-01
4157	2.091058E-01	0.000000E+00	2.091058E-01	-1.788427E+01	-2.794257E-01	3.084500E-01
4158	1.199914E+00	0.000000E+00	1.199914E+00	-1.479647E+01	-6.026961E-01	7.571209E-01
4159	2.357198E-01	0.000000E+00	2.357198E-01	-1.790755E+01	-2.221651E-01	3.270921E-01
400000	5.531502E-01	0.000000E+00	5.531502E-01	-1.655866E+01	-1.286234E-01	1.278799E+00
400001	6.054949E-01	0.000000E+00	6.054949E-01	-1.749011E+01	5.523733E-01	3.457566E-02
Total :	3.388149E+02	0.000000E+00	3.388149E+02	-1.787833E+01	6.316578E-02	5.718306E-03
Average :	N/A	N/A	N/A	N/A	N/A	N/A

9.8.5. [WRITE] COINC_ELEMS Output of Coincident Element Lists

[WRITE] COINC_ELEMS Output of coincident element lists

A "coincident" element is one that shares the same topology as another element of the same type, but not necessarily in the same order. Coincident elements of different types, for example contact segments on shells, do not qualify.

There are no further arguments or sub-menus for this command, and an example listing is:



Entity ID	Element ID	Element ID
H600379	H600386	N/A
H600774	H600781	N/A
S801179	S804496	N/A
S801181	S804497	N/A
S801182	S804498	N/A
S801342	S804508	N/A
S801343	S804509	N/A
S801361	S804516	N/A
S801363	S804517	N/A
S801364	S804518	N/A
S801377	S804528	N/A
S801378	S804529	N/A
SP1550001	SP1550002	SP1550003
SP1550005	SP1550006	N/A
SP1550007	SP1550008	N/A
LM400000	LM400021	N/A
LM400001	LM400022	N/A

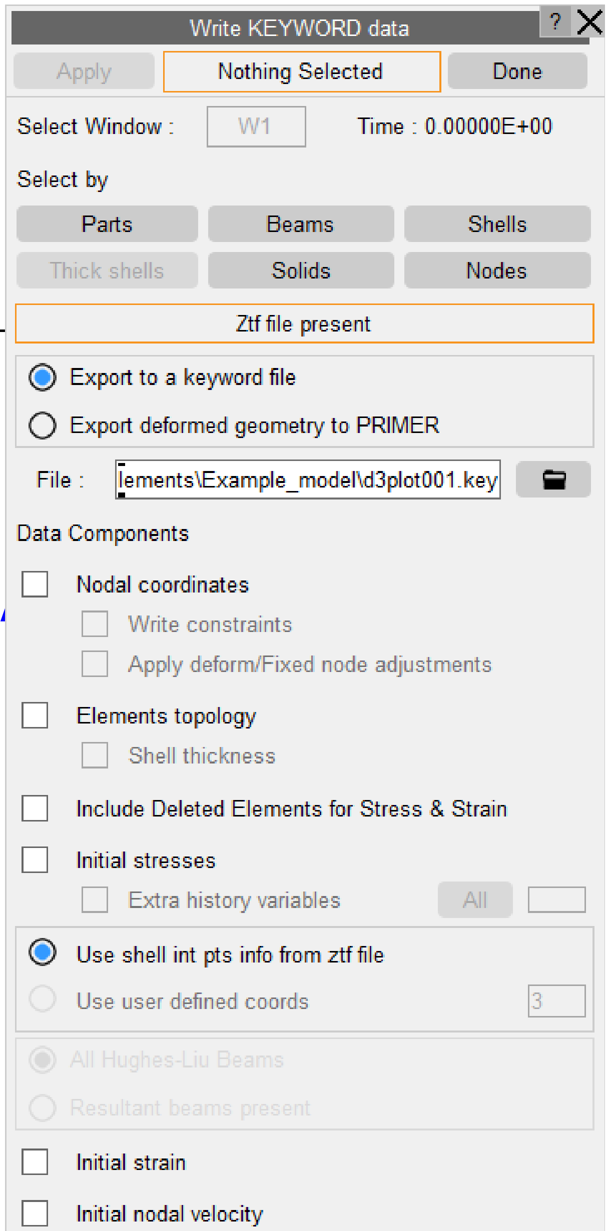
9.8.6. [WRITE] UNATT_NODES Lists of Unattached (Non-Structural) Nodes

[WRITE] UNATT_NODES Lists of unattached (non-structural) nodes

D3PLOT considers a node to be "unattached" if it is not part of the topology list of any element. It does not, for example, know about "extra nodes on rigid bodies", so many nodes may get listed as unattached when in fact they are restrained in some way.

A typical listing is:

```
Unattached nodes
--------------------------------------------------------------------------
------
3007      4007     6005     10002
10008     10009    10010    10011
10018     10019    10020    10021
10502     10503    10505    10506
10511     10512    10514    10515
10521     10522    10523    10524
11004     11005    11006    11007
11012     11013    11014    11015
11020     11021    11022    11023
--------------------------------------
------
```



9.8.7. [WRITE] KEYWORD DATA

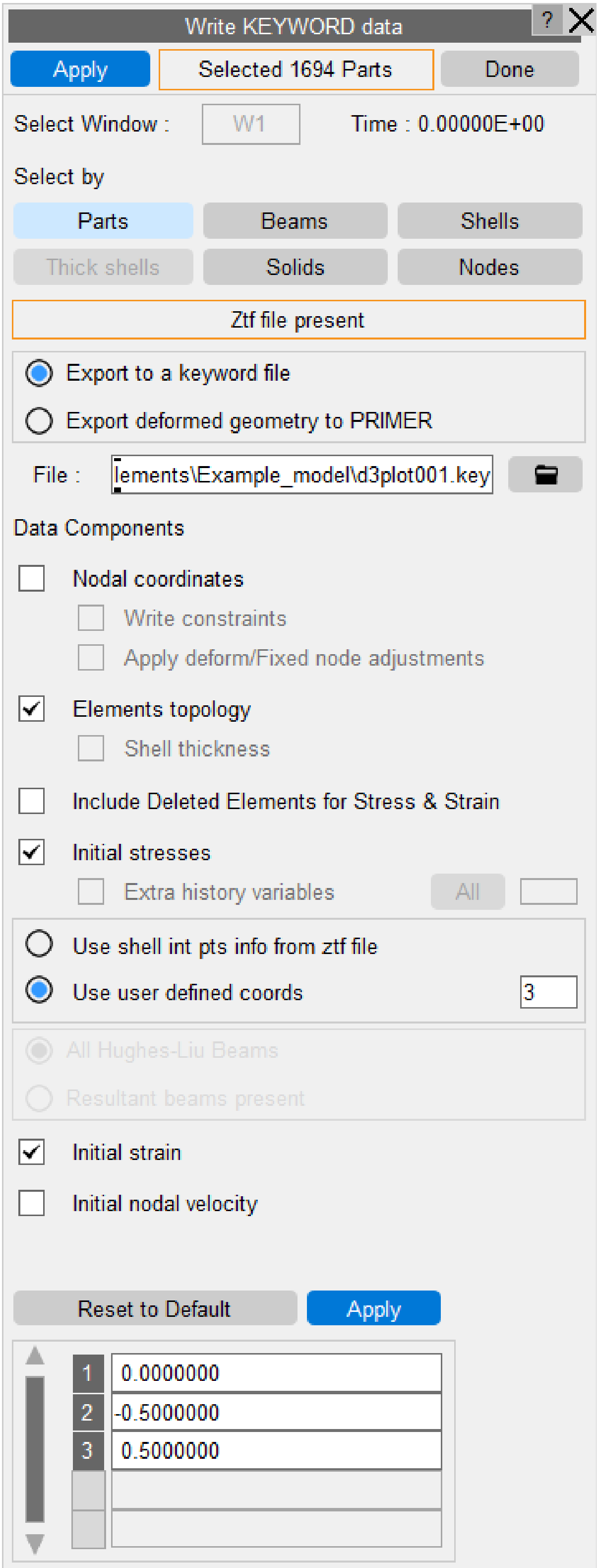
[WRITE] KEYWORD DATA

This figure (right) shows the **KEYWORD DAT**

"keyword data" output means results for a <list> of standard entity types written into a keyword file instead of output to screen. The keyword output can also be exported back to the original keyword model in an integrated PRIMER session. The output is similar to that of [WRITE] ENTITY, but it has been arranged into DYNA keyword format so that it can be read into DYNA as initial data of a model.

To use this panel:

- Select if the keyword data needs to exported to PRIMER or written out into a file
- Select an entity type, eg **Beams** or **Shells**
- Define a <list> of entities in the selection popup box
- Select the data component(s)
- Select the output .key file OR export to PRIMER
- Press **APPLY** to generate output OR export to linked PRIMER session



9.8.7.1. What Does a "Keyword Data" Output Produce?

What does a "keyword data" output produce?

You can write out five properties in keyword format about any entity in your database. One example with 1694 parts selected for output is shown below.

Select By

You can select the elements to write output for using the "Parts", "Beams", etc buttons at the top of the menu. A list of all the entities will be available to select in the popup menu. In our example, shells were required, so "Shells" was selected.

ELEMENT TOPOLOGY

Selecting "Elements topology" will write out the *ELEMENT cards for the selected elements.

For shell elements, if thickness data is present in the results (aka. if the results contain shell thickness change for deformable shells [ISTUPD option in *CONTROL_SHELL]), you can also tick "Shell thickness", in which case *ELEMENT_SHELL_THICKNESS cards will be written with the shell thickness at the selected state. This option does not write the BETA / MCID paramater to the card, as this information is not available to D3PLOT.

INCLUDE DELETED ELEMENTS for Stress & Strain

If this option is selected, when INITIAL STRESS or INITIAL STRAIN (see below) is selected to be exported, initial stress and initial strain of deleted elements will be exported. Otherwise, initial stress and initial strain of deleted elements are excluded from the export.

INITIAL STRESS and INITIAL STRAIN

Special attention is given to *INITIAL_STRESS_SHELL/TSHELL and *INITIAL_STRESS_BEAM as they both can write information about more than one integration point of the element chosen. The ztf output from PRIMER will include the information of integration points for any shell and beam included in the model. From the ztf file D3PLOT will look into *SECTION_SHELL/TSHELL, *INTEGRATION_SHELL and *PART_COMPOSITE for shells' integration point information and *INTEGRATION_BEAM and *SECTION_BEAM for beams' integration point information. Combined with the data at these integration points contained within the ptf file, we are able to write full initial keywords.

When a ztf file is not present, user defined number of integration points of shells and their coordinates can be set as shown in the figure on the right. In this case, the number of integration points is set as MAXINT from the CONTROL Card in ptf files at default, which is the number of integration points the ptf file has data for. By selecting to "Use user defined coordinates" the editing panel for the coordinates will appear at the bottom of the WRITE panel. The values in that are decided using Gaussian integration rule with the specified number of integration points. You can edit the values by Apply or Reset to default values. In the example, Gaussian rule cooordinates for 3 integration points have been edited from 0, -.7745967, +.7745967 to 0, -0.5, 0.5.

Only Hughes-Liu integration beams can have cross section integration with quadrature integration rules. With a ztf file D3PLOT can find this information itself so the option is always greyed out. However, without a ztf file you need to decide whether they are all Hughes-Liu beams in which case the number of integration points will be decided by the BEAMIP for beams in the Control Card in the ptf file. If "Resultant beams present" is selected, it'll assume there's only one integration point in all the beams.

There are cases where BEAMIP and MAXINT is set zero in the model which leads to no stress information for Beams or Shells respectively, D3PLOT therefore will not write *INITIAL_STRESS_BEAM or *INITIAL_STRESS_SHELL.

If extra variables have been output to the ptf file these can be written to the *INITIAL_STRESS cards. When this option is ticked the number of extra variables to write defaults to all the variables that have been written to the ptf file. This number can be reduced manually by entering a new number in the text box, for example if writing intial stresses for parts of a material that does not use as many extra variables as have been written, or reset to all of the extra variables by pressing the "All" button.

NODAL COORDINATES and INITIAL NODAL VELOCITY

When nodal information of elements is required, such as Nodal coordinates (*NODE) and Initial nodal velocity of shell (*INITIAL_VELOCITY_NODE), beam and solids, for the elements selected, the nodes on such elements will be automatically flagged for nodel information output. Please note that by selecting all Entities, it will only include nodes on all Entities, this does not guarantee all nodes in the whole model to be included for information output.

A switch is available to select whether or not to write the nodal restraints to the *NODE card. This can be useful if you want to merge the exported nodes back into a model without the original constraints.

Relative nodal coordinates also can be written out using this panel. Deform-> Fix node or Deform -> Shift def panel can be used to turn on the relative nodal coordinates. To write out the current relative nodal coordinates turn on **Apply deform/Fixed node adjustments** .

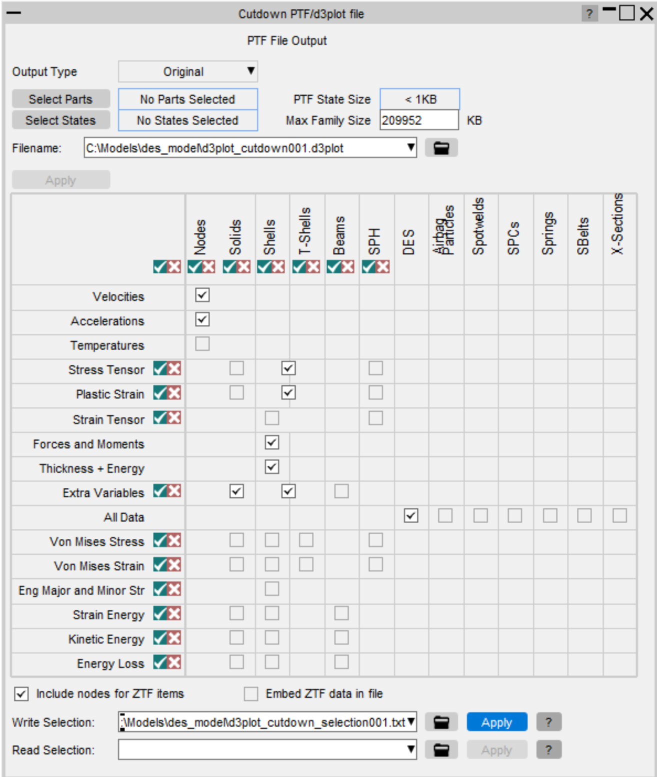
```
*KEYWORD
$ Keyword file written by D3PLOT from plot file
$ Z:\testfiles\cantilever\beam_shell.ptf
$
$
$ ++++++++ Data at time 0.20008E-02 ++++++++
*INITIAL_STRESS_BEAM

        12                    4          1
          2

  3.594E+01  0.000E+00  0.000E+00  0.000E+00  0.000E+00  1.994E-01   0.000E+00
  3.594E+01  0.000E+00  0.000E+00  0.000E+00  0.000E+00  1.994E-01   0.000E+00
 -3.592E+01  0.000E+00  0.000E+00  0.000E+00  0.000E+00  2.761E-01   0.000E+00
 -3.592E+01  0.000E+00  0.000E+00  0.000E+00  0.000E+00  2.761E-01   0.000E+00


*INITIAL_STRESS_SHELL

        31                  3        0         0
          1

 0.0000000  3.479E+00 -3.919E-  4.838E-   3.005E-  4.622E-   4.487E- 0.000E+00
                           04       02        07       06        01
-0.5000000 -1.094E-02 -7.716E-  2.783E-  -5.075E-  4.560E-   1.809E- 0.000E+00
                           07       02        07       06        01
```

```
 0.5000000 -3.501E+00  3.903E-  7.282E-  -1.316E-  4.498E-  -8.687E- 0.000E+00
                            04       03        06       06        02

*INITIAL_STRAIN_SHELL
          31
 -2.756E-07  -7.238E-08 4.445E-07  -9.425E-12 8.468E-11 1.994E-01   3.360E-06
 -5.005E-05   1.498E-05 1.511E-05  -2.449E-11 8.357E-11 1.994E-01  -1.613E-06

*ELEMENT_BEAM
    12           100     101    127
      8


*ELEMENT_SHELL
    31            80      81      78     206
      1


*NODE
       78 8.989016E+01      1.500000E+02    3.950482E+00         0.         0.
       80 9.986085E+01      1.600000E+02    4.715574E+00         0.         0.
       81 8.989016E+01      1.600000E+02    3.950481E+00         0.         0.
      100 9.999877E+00      2.500000E+02    4.969115E-02         0.         0.
      101 1.999884E+01      2.500000E+02    1.934823E-01         0.         0.
      206 9.986085E+01      1.500000E+02    4.715574E+00         0.         0.

*INITIAL_VELOCITY_NODE
  78   -2.744E+02   8.881E-04   5.102E+03
  80   -3.398E+02  -7.816E-05   5.955E+03  0.000E+00  0.000E+00   0.000E+00
  81   -2.744E+02  -9.552E-04   5.102E+03  0.000E+00  0.000E+00   0.000E+00
 100   -2.869E-01   0.000E+00   5.765E+01  0.000E+00  0.000E+00   0.000E+00
 101   -2.618E+00   0.000E+00   2.204E+02  0.000E+00  0.000E+00   0.000E+00
 206   -3.398E+02   5.687E-05   5.955E+03  0.000E+00  0.000E+00   0.000E+00
```

9.8.8. [WRITE] CUTDOWN D3PLOT/PTF File

[WRITE] Cutdown D3PLOT/PTF file

This figure (right) shows the menu for writing a compressed set of PTF files. There are two output options available:



- ORIGINAL - the data is written to the file in the normal format (it will be able to be read by other post-processing software)
- REORDERED - the file is written in a format that makes it quicker for D3PLOT to read (it will only be able to be read by D3Plot)

This option can be used to generate a new set of PTF/D3PLOT files for a model that contain only a subset of the Parts and States in the original model.

As well as reducing the size of the new PTF/D3PLOT files by reducing the number of PARTs and STATEs it is also possible to remove some of the data components from the files.

If the file is written in the REORDERED format derived components Von Mises Stress, Von Mises Strain and Engineering Major and Minor Strains can be embedded in the file. If the stress/strain tensors used to derive them are not needed they can be omitted from the file to save disk space.

LSDA (binout) data components can also be embedded in the REORDERED format file. This could be useful if the LSDA file is large as the data components cannot be plotted until it has been read in. By embedding them in the file they are available instantly.

This operation can also be performed from the dialogue interface and in batch using the following methods:

Using Dialogue prompt commands	**/UTILITES PTF_COMPRESS**	See [Appendix E](#) for the full set of commands.
Using the command line	**-ptfcut=** *<filename>*	

(These commands may be used in non-graphical mode using the "tty" display device, making them suitable for use on a typical remote cluster.)	Where <*filename*> contains a list of /UTILITIES PTF_COMPRESS commands, equivalent to typing these at the command prompt	See CLINE_PTFCUT for more details
	`-ptfcut_batch` < *further options* > **`-o=`** < *output_filename* > This permits common cases (eg just strains for all states) to be output without having to define a file of commands	

using the dialogue commands:

OUTPUT TYPE	Select whether to write out the file in ORIGINAL or REORDERED format.
SELECT PARTS	This option can be used to select a subset of the model. After selecting the PARTs that are going to be written to the new set of PTF files the **PTF State Size** will be recalculated to show the size of each PTF state in the new files.
SELECT STATES	This option can be used to select a subset of the STATEs in the original model for output to the new set of PTF files.
MAX FAMILY SIZE	The maximum family member size that will be written out can be specified here. By setting this high enough it can be used to fit all the states into one file. By default it will be set to the size of the maximum family member in the model that is being cut down. NOTE: If the family size is set to a value lower than the state size, when D3Plot writes out the file it will silently increase the family size so that it will create one state per family member.

As well as selecting a subset of PARTs and STATEs it is also possible to turn off the output of some data components to the new PTF files. Depending on what data componets and entity types the original model contains the table in the bottom section of the menu can be used to either select or deselect optional data components for output.

Due to the way the PTF file format works it is not always possible to turn a data component on/off for a single entity type. If for example your model contains both Shells and Thick Shells then both the **`Stress Tensor`** and **`Plastic Strain`** must be either be truned on or off for both types of elements. Similarly the **`Strain Tensor`** must apply to all Solids, Shells and Thick Shells.

As data components are turned on/off the **PTF State Size** will be recalculated to show the size of each PTF state in the new files.

INCLUDE NODES FOR ZTF ITEMS

The PTF/D3PLOT files only contain information for the basic LS-DYNA element types. If you also have a ZTF or XTF file then D3PLOT will use the extra information in these files to draw Springs. Joints etc.

If you select a subset of PARTs for output it is possible to select a PART that is attached to one end of a Spring without selecting the PART connected to the other end of the spring. If this option is selected then D3PLOT will automatically ensure that the nodes at the end of any springs (or any other elements) that are no-longer attached to PARTs are also output to the new PTF files so the items can be drawn correctly.

EMBED ZTF DATA IN FILE

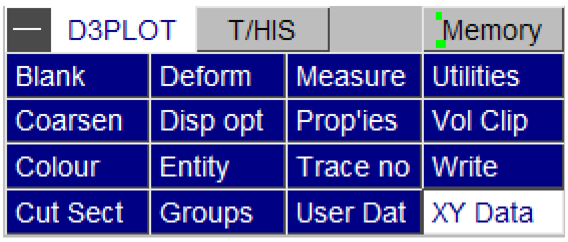
If the file is being written in REORDERED format you can embed the ZTF data into the file.

WRITE SELECTION / READ SELECTION

These two options can be used to save and retrieve the options selected in the panel by writing a file containing the relevant dialogue commands. This file can be used as an argument on the command line (-ptfcut=<filename>) to ouput a cutdown version of the ptf file. This could be done as part of an automatic process at the end of an analysis to create files with a subset of results and/or reordered to make post-processing faster.

9.9. XY_DATA Drawing Numerical Data as XY Plots and/or Writing it to File

XY_DATA Drawing numerical data as XY plots and/or writing it to file



The **XY_DATA** command lets you extract scalar data values in a way similar to **WRITE** , but in this case over a range of times, and to plot them and/or write them to file.

XY_DATA is similar in principle to **TH_DATA** , the T/HIS link, but it operates on low-frequency data extracted from the plot database files (.ptf etc) rather than high frequency data from the time-history ones (.xtf etc). The **TH_DATA** command is described in T/HIS the D3PLOT <=> T/HIS link

Extracted values are buffered in backing store, and remain there available for subsequent replotting until superseded by further data extraction. This backing store is limited in size, but it should be adequate to cope with extracting data for at least 500,000 items.

9.9.1. Using the XY_DATA Commands

Using the XY_DATA commands

The **XY_DATA** main menu is shown right.

Select States	Defines the time states used for output. (By default all states are preselected)

Three different data extraction and plotting methods are provided, and are descrived in more detail below:

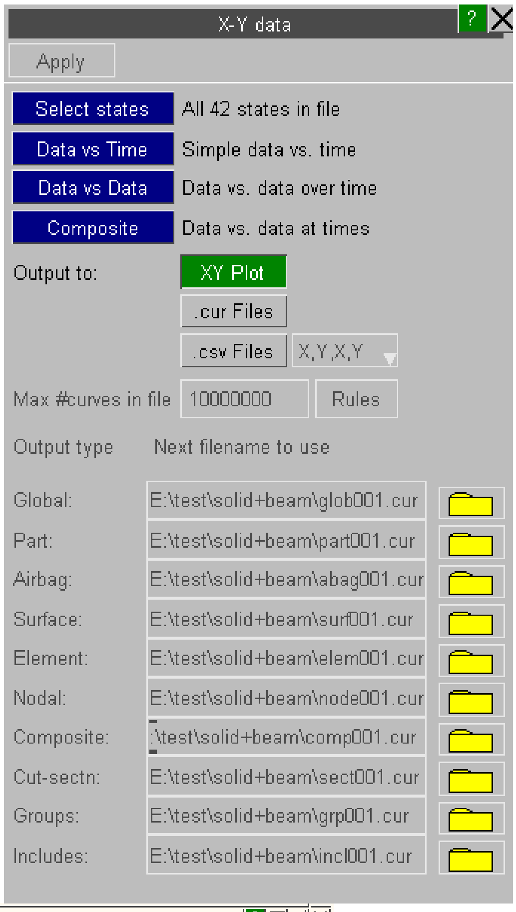
(1)	**Data vs Time**	Plots data (Y axis) vs time (X axis)
(2)	**Data vs Data**	Plots data (X and Y axes) over a range of times
(3)	**Composite**	Plots data (X and Y axes) for a range of items at a time.

Output may be sent to any of:

XY Plot	On screen graphical tool that displays curves. It permits curve selection, zooming, labelling and symbol display; it also allows you to select a subset of curves to be written to file.
.cur Files	ASCII format files written in "curve file" (.cur) format. The default filenames will become "live" if you select this option, and you may use these or define your own names.
.csv Files	Comma Separated Value format files. The curves may be written in "X,Y,X,Y" or "X,Y,Y,Y" format.

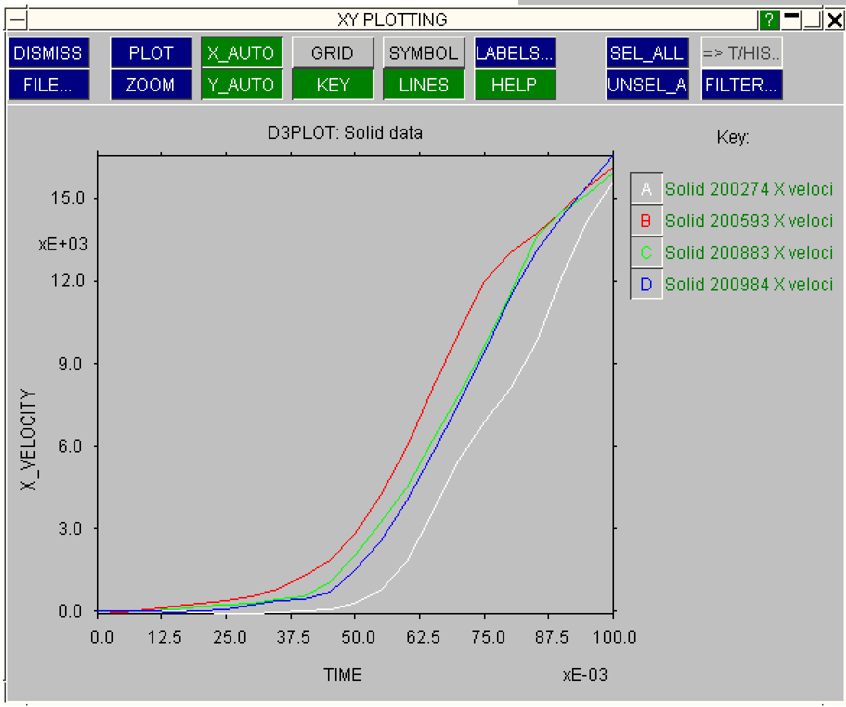
The default filenames will become "live" if you select this option, and you may use these or define your own names.

The data requested are extracted from the complete states as they are read in, and stored on scratch backing store. When everything has been read in then the file(s) are generated and the XY plot drawn according to what output has been requested. XY plot management is described in Using the XY graphical plotting tool, and file management in Managing "Curve" File Output.



The data on backing store is "remembered" until you leave the programme, or overwrite it by reading in some more results. So once a set of data has been created you can exit and re-enter **XY_PLOT** at will to review it.

Typical graphical output from **XY_PLOT** is shown in the figure below:



A typical file (the results for Material 1 above) is shown below. This is written in T/HIS "Curve" file format (see Section 4.8.5.5 for a description of this format):

```
$ ---------------------------------------------------------
$ D3PLOT "T/HIS_INTERFACE" output file. (22-Nov-95 20:22:11)
$ Database file: /users/dyna70/test/lg09
$ Title: lg09 : Large Test 9: Belted sled test
$ ---------------------------------------------------------


D3PLOT: Material data
TIME
KE_KINETIC_ENERGY

Material 1          Kinetic energy
.000000E+00         7.222228E-05
2.490001E-03        1.272031E+01
4.990020E-03        3.979501E+01
7.490039E-03        6.687062E+01
9.999956E-03        9.656306E+01
1.249986E-02        1.276856E+02
1.499976E-02        1.723507E+02
1.749984E-02        2.691978E+02
1.999997E-02        4.552962E+02
2.249011E-02
                    7.637439E+02

2.499991E-02
                    1.173867E+03
```

9.9.2. SELECT_STATES Choosing the Complete States to be Used for Output

SELECT_STATES Choosing the complete states to be used for output

The figure (right) shows the **SELECT_STATES** control panel.

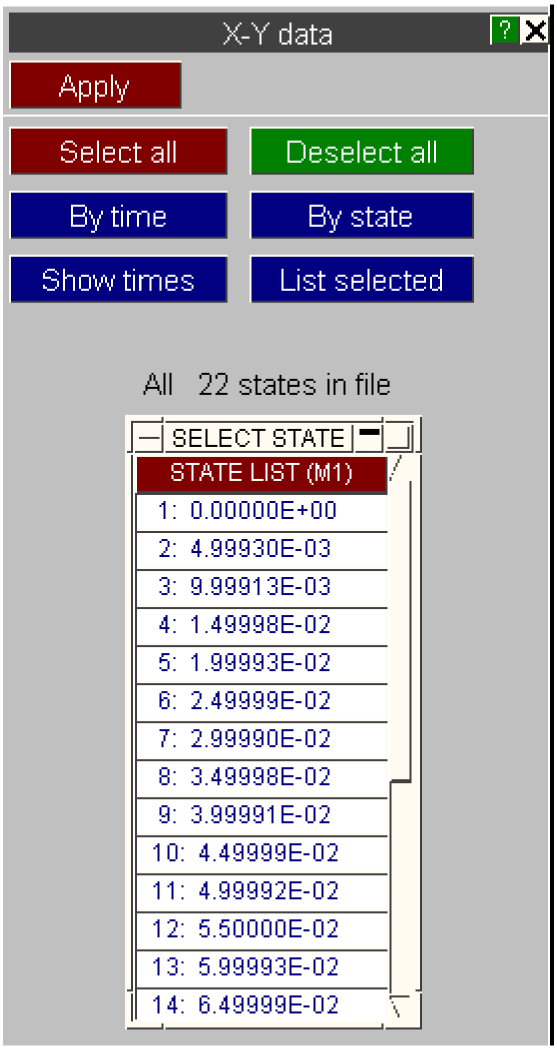
By default all states in the file are selected for **XY_PLOT** output, but you may select any sub-set that you wish.

Once your selection is complete use **APPLY** to return to the main **XY_PLOT** menu.

The commands to select and modify the states selected for **XY_PLOT** output are identical to those used for animation. (Ref ANIM > Controlling the animation process .)

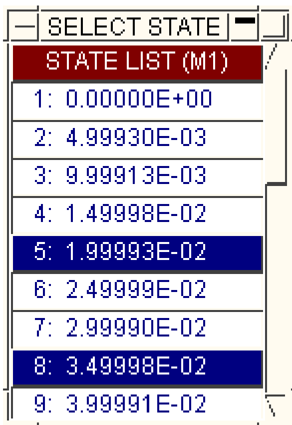
However the internal storage of selected state numbers is totally separate: states selected for XY plotting will not affect those stored for animation, and vice-versa.

States may be selected using **BY_TIME** (start rime, interval, end time) or **BY_STATE** (start state, interval, end state)



It is also possible to select and deselect times from the **STATE_LIST** menu. Click on an undepressed state to select it, or on a depressed state to deselect it.

This menu always the shows what has been selected, regardless of the selection method used. And you can mix selection methods: for example you could select states by time, then modify the result in this menu.



Important notes on internal state lists:

Order is important:	The order in which states are selected is significant. If you pick entries #1, #4, #2 from the menu here, or when defining a <list>, then that is the order in which they will be processed. Implicit in this is the fact that states can be repeated, for example the sequence <#1, #2, #3, #2, #1> is quite legal, subject to the space restrictions below.
Space is limited:	The amount of space set aside for storing state lists is 2x the number of states available. Thus you could store the sequence <#1 - #n - #1>, but no more than this.

9.9.3. Data vs Time Generating Data vs Time

Data vs Time Generating data vs time

This is the simplest form of XY plotting:

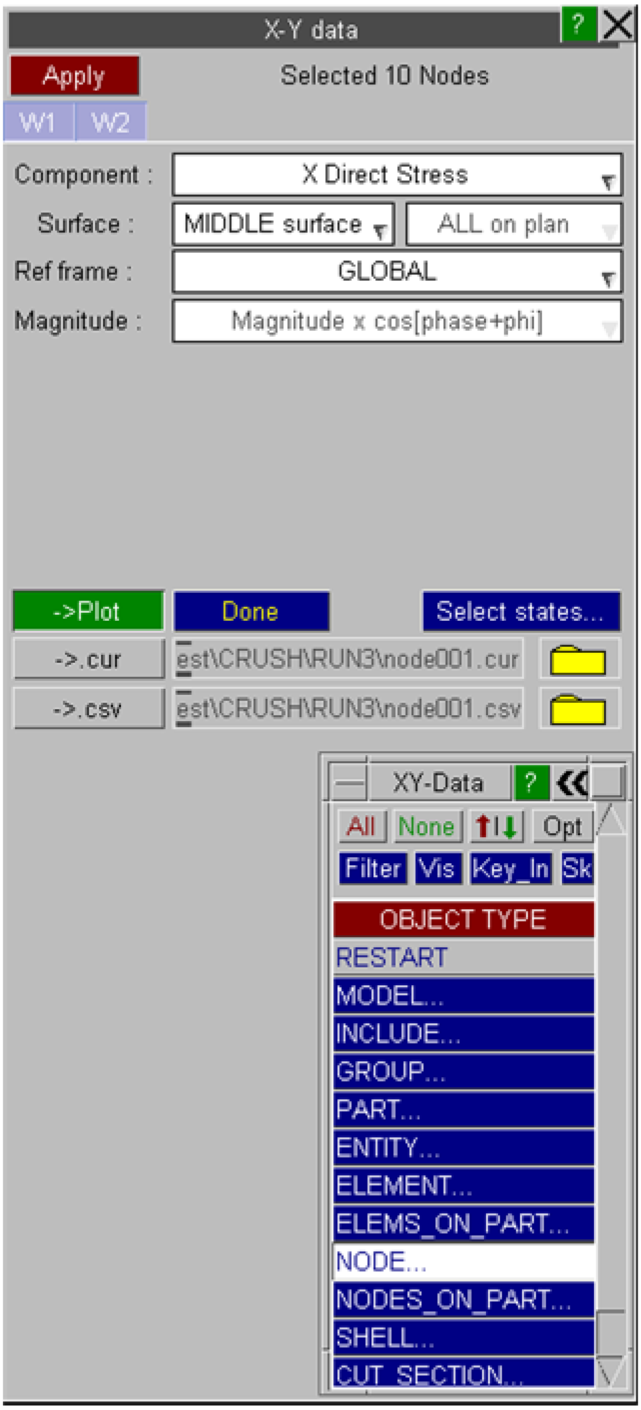
- You select a list of items (here shells have been chosen)
- You select a data component (here X Direct Stress)
- Optionally select a shell surface using the left hand integration point drop down menu
- Optionally select shell on plan integration point(s) / solid integration point(s) using the right hand integration point drop down menu
- Optionally select a frame of reference

Apply will then extract and plot the data.

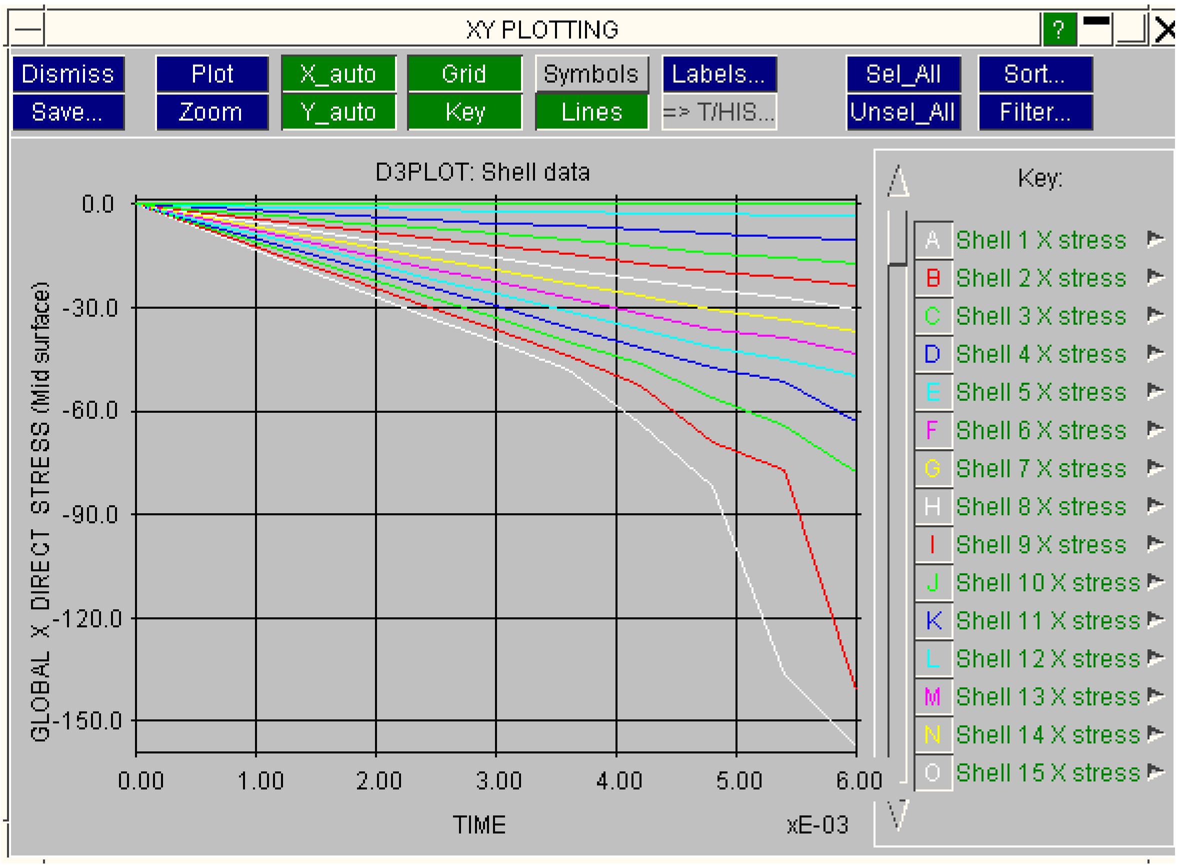
For each item chosen, here shells, a curve will be generated of

X axis: Time
Y axis: Selected data component

-> Plot	Toggles display to the graphical XY plotting tool
-> File	Toggles output to a curve file, and you can define the filename.
Select states	Allows you to define different states for output
Done	Returns to the main XY plotting menu



Here is the output generated by the panel above:



9.9.4. Data vs Data Both X and Y Data Components Against Time

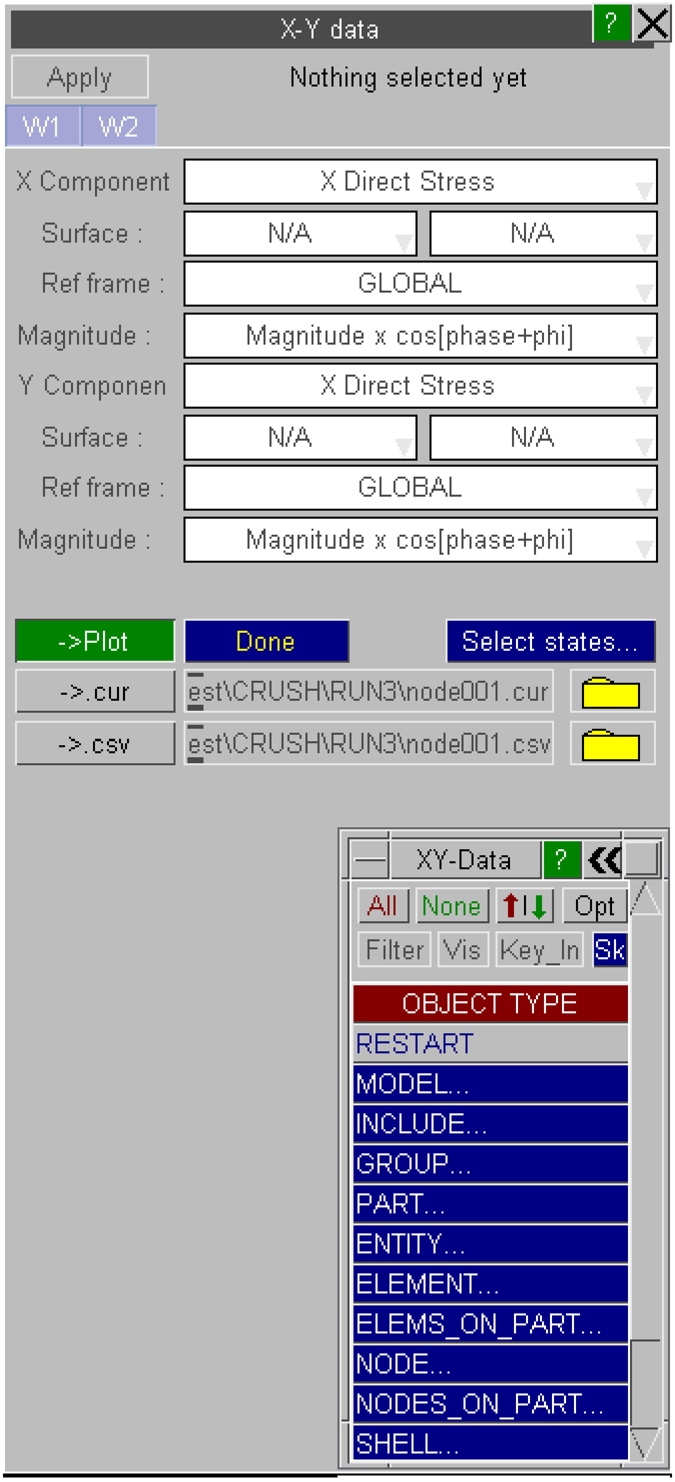
Data vs Data: Both X and Y data components against time

This is very similar to the Data vs Time panel above, except that you choose data components for both X and Y axes.

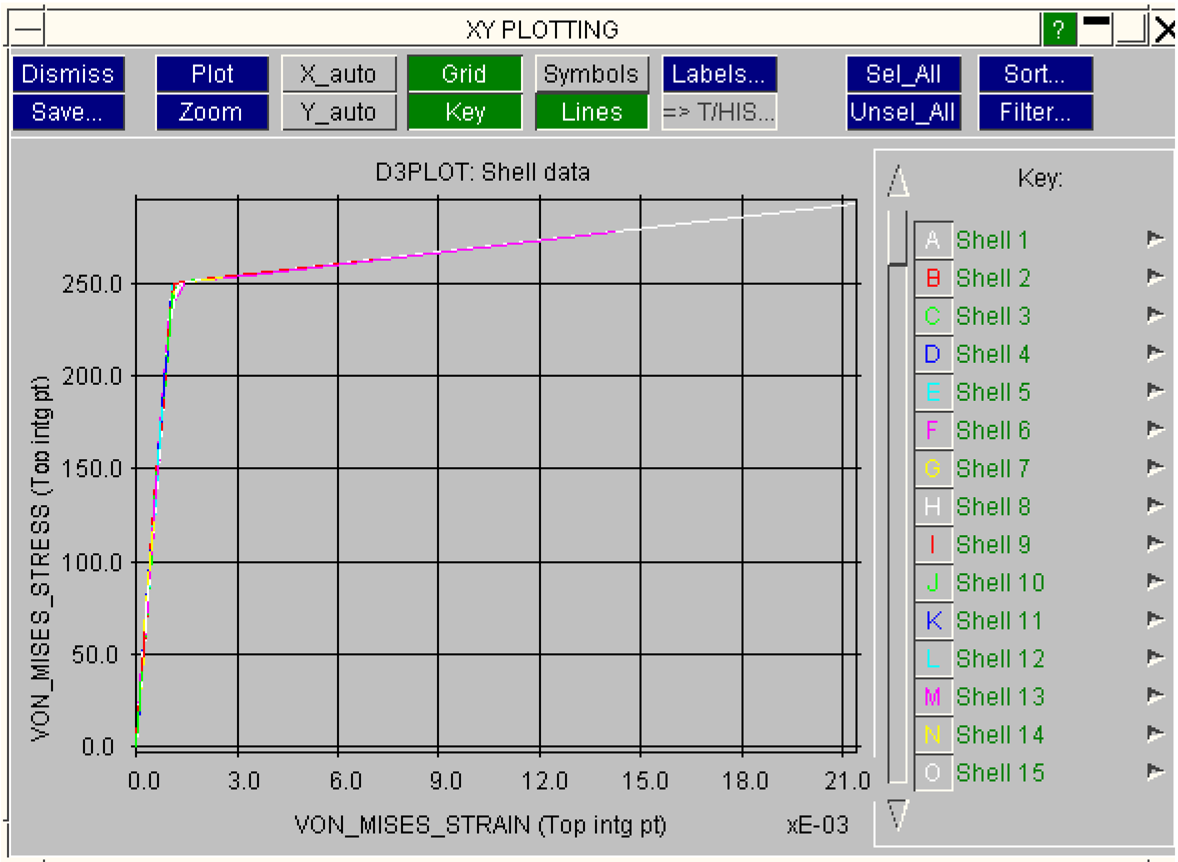
Each item (here shells) still forms a curve, but the data points are <X data> vs <Y data> at each selected state.

For example you might wish to plot the stress vs strain history for a set of elements, in which case you might choose:

- X Axis: Strain
- Y Axis: Von Mises stress



Here is an example of output from such a plot, showing Von Mises Strain (X axis) against Von Mises Stress (Yaxis).
This shows the classic stress vs strain curve for mild steel.



9.9.5. COMPOSITE_DATA Data vs Data for a List of Items Over a Range of Times

COMPOSITE_DATA Data vs data for a list of items over a range of times

"Composite" data is similar to "Data vs Data" except for the vital difference that:

- Each curves contains (x,y) data for the list of items selected, in order, at a particular time.
- Therefore the number of curves = the number of states selected, not the number of items
- And the number of points in each curve = the number of items selected.

This plotting mode is useful for displaying the variation of a quantity across space in a model over a range of times, as the examples below show.

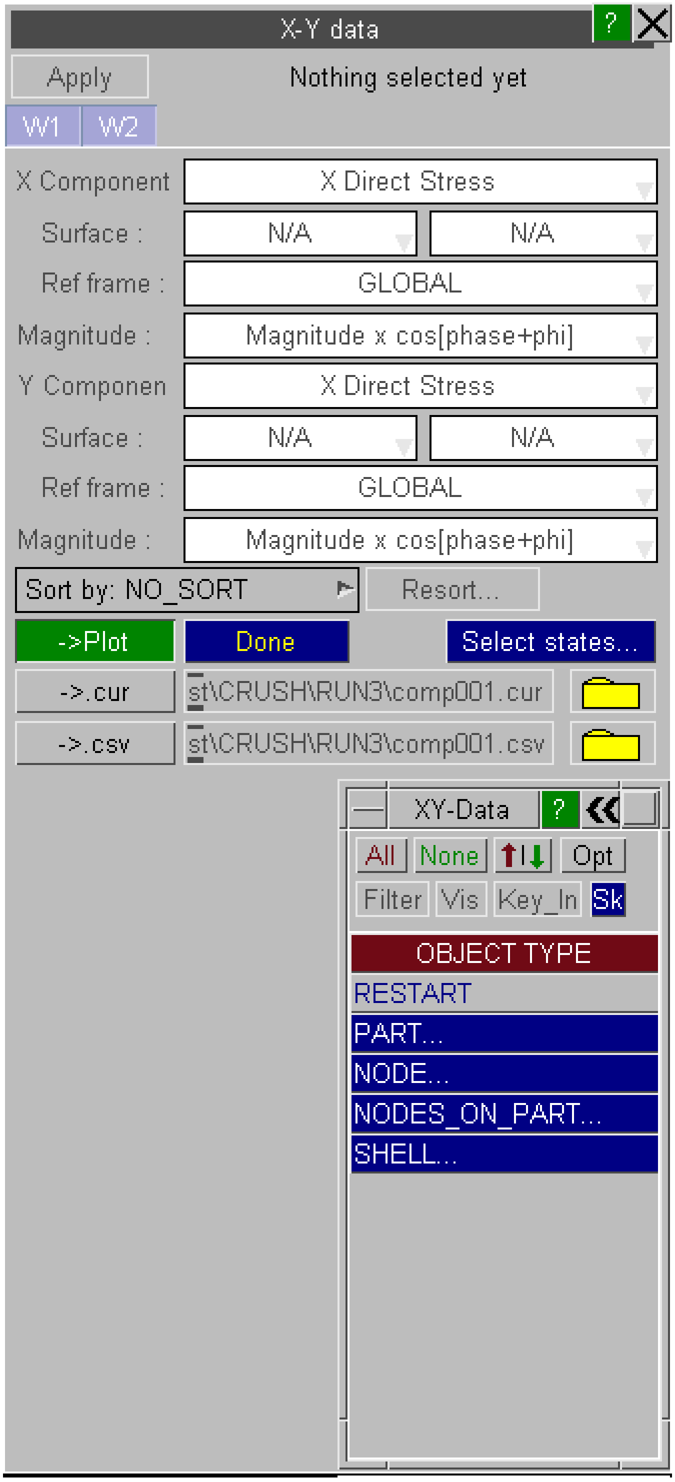
IMPORTANT: The order in which points are written in composite curve is the order in which they are defined, unless subsequently sorted.

- If you screen-pick items they will appear in the order picked up to a limit of 1000 items

- If you select by area, polygon on "all" they will appear in ascending label order

Sort by: sorting data points.

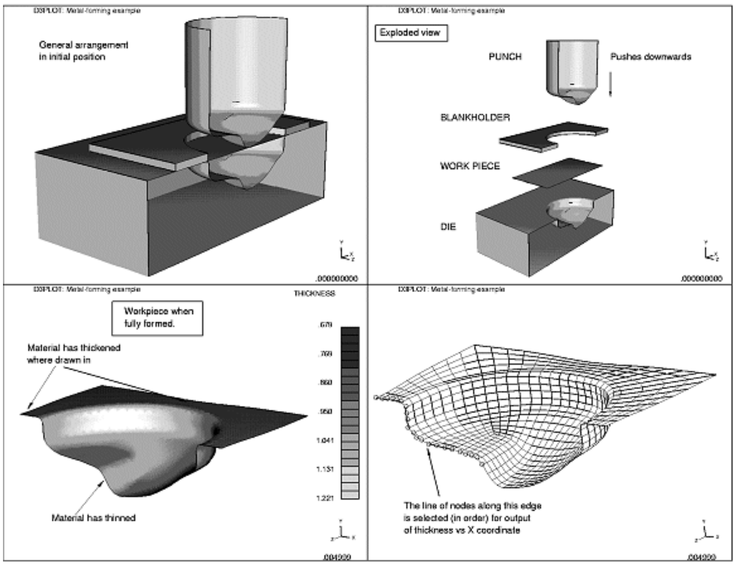
When points are picked by screen area, or some other unordered means, their order is usually "wrong". You can use the **Sort by** options to correct this, and these are described in Sort By: Sorting Data Points in Composite Plots.



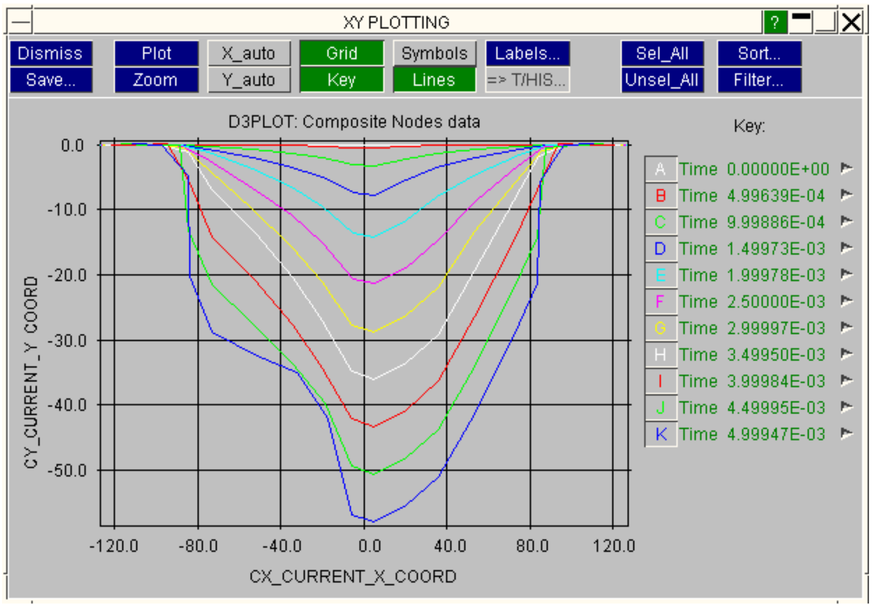
9.9.5.1. Example: Displaying Profile Change During Metal Forming

Example: Displaying profile change during metal-forming

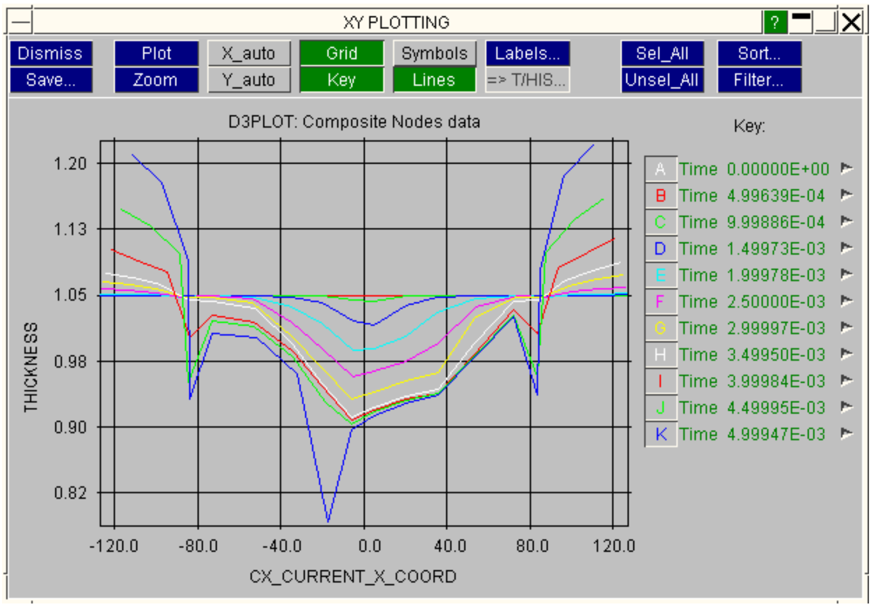
The figures below show a metal-forming operation, and the way composite plots could be used to interpret the results. Clearly the order in which the list of nodes (along the edge of the workpiece) is defined is important for the composite plots below.



Composite plot of X vs Y coordinates showing formed profile



Here is a composite plot of thickness vs X coordinate for the same model

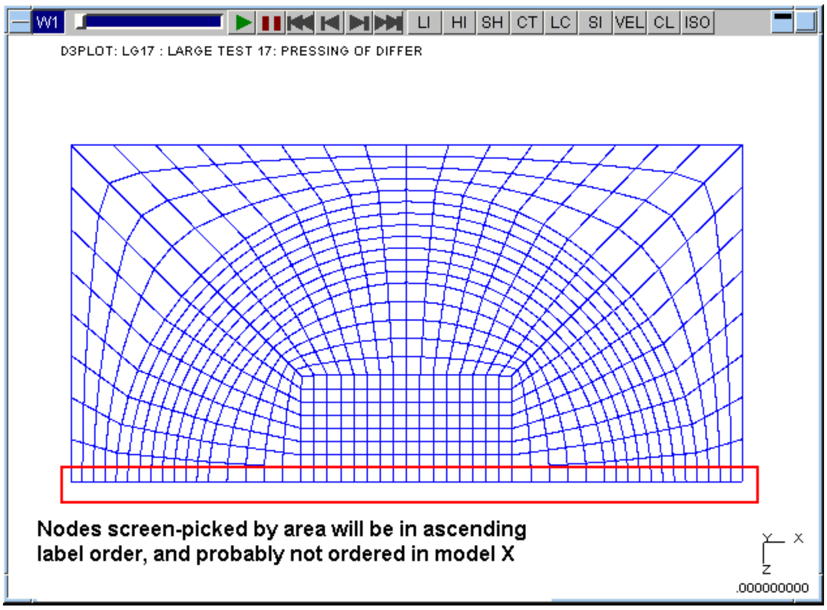


9.9.5.2. Sort By: Sorting Data Points in Composite Plots

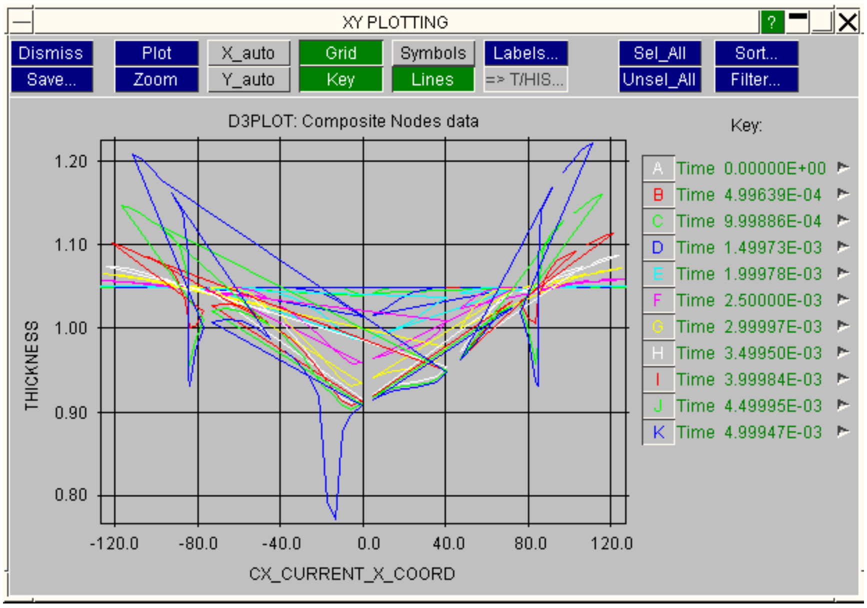
Sort By: Sorting data points in Composite plots

In the examples above nodes along the workpiece edge were selected carefully, and in order, by screen picking. This can be time-consuming so an alternative is to select by area and then to sort by some criterion. The example below shows what may happen, and how to correct it.

Screen-picking nodes by area



Sure enough, some of the nodes are not ordered in ascending X coordinate, giving a muddled plot.



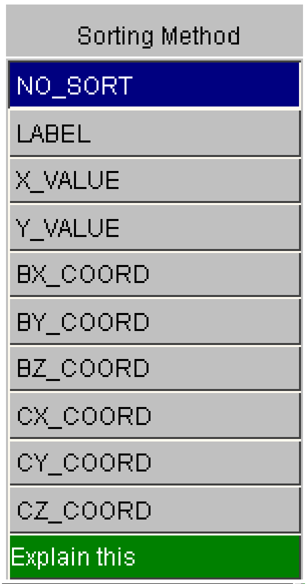
By default no sorting takes place, but you can sort by any of the options in this menu.

In this case sorting by **CX_COORD** , the current X coordinate, will work for the plot above, and the plot below shows how this corrects the image.

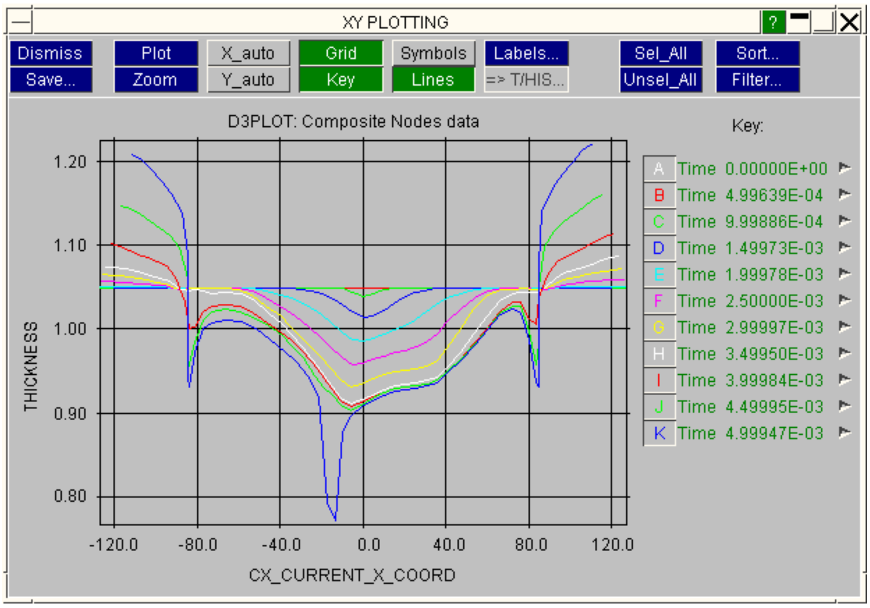
Note that:

- The current **Sort by** option remains current until changed. Any data subsequently extracted will be sorted in this way.

- You can resort data currently cached in backing store at any time



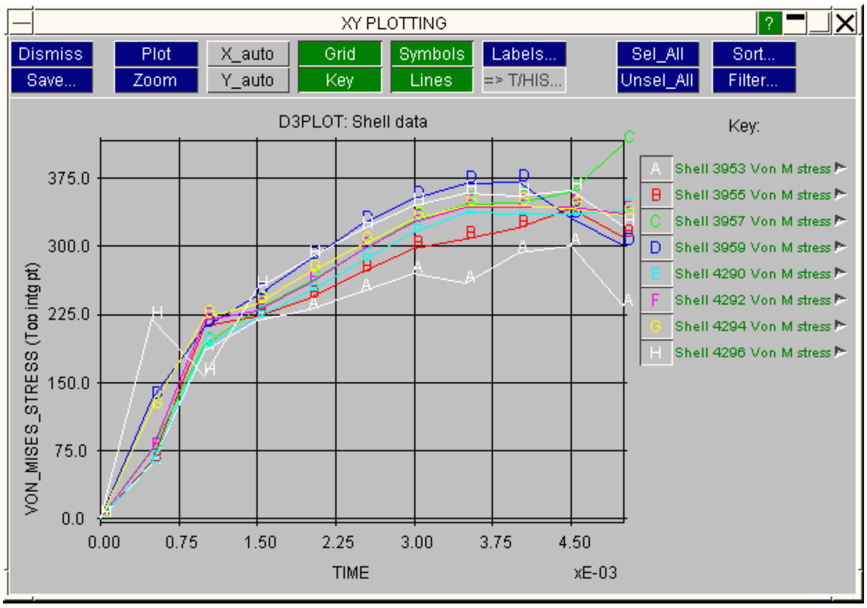
After sorting by Current X coordinate the points are back in a sensible order.



9.9.6. Using the XY Graphical Plotting Tool

Using the XY graphical plotting tool

When XY plots are requested they are drawn using a standard XY plotting tool. This operates using a menu system window, so it is not available if the screen menu interface is not being used. The figure below shows a typical XY data plot.



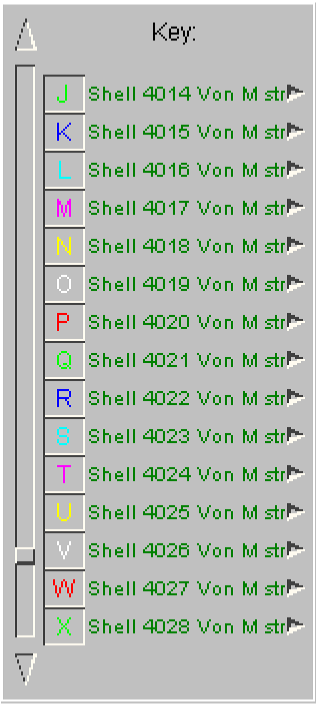
9.9.6.1. Selecting the Curves to be Plotted

Selecting the curves to be plotted

By default all curves are plotted, but you can select and deselect individual curves by clicking on their label buttons (under **Key:**). If there are too many curves to fit into the vertical space availble a scroll-bar is added as shown to the right.

Only those curves selected for plotting are drawn when the next **PLOT** command is issued. You can use **SEL_ALL** and **UNSEL_ALL** to (de)select all of them, and the **FILTER** options to select and deselect curves by colour and/or label: see below.

The colours (white, **red** , **green** , blue, **cyan** , **magenta** , **yellow**) and letters (A-Z) of curves are fixed and cannot currently be changed. The letters are used as "symbols" when symbol display is sturned on, as shown in the plot above.



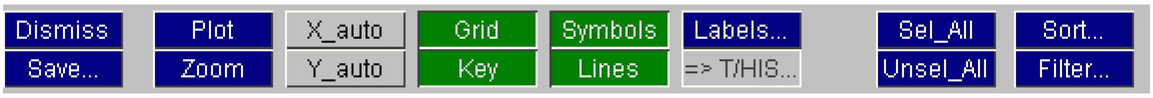
Obtaining information about curves

You can use the popup menus against any "live" curve in the scrolling key to obtain the max/min X and Y axis data for that curve.

Here curve **K** has been selected.

9.9.6.2. Top Commands

Top commands



Dismiss Closes the XY plotting window. Data is left unchanged on backing store.

Save... Writes the currently selected curves only to "curve" or "csv" files. The syntax and other conventions for these files are described in [Managing "curve" file output] .

Plot Redraws the current graph with only those curves selected for plotting.

Zoom Uses the cursor to zoom in on a rectangular area of the graph, and enlarges this to fill the whole window. Both X and Y scales become set explicitly, ie taken out of automatic mode, by this command.

X_auto Resets the X scaling to automatic.

Y_auto Resets the Y scaling to automatic.

Grid Toggles the grid at tick mark intervals on/off. (Default ***off***)

Key Toggles the curve "Key" listing on/off. (Default on)

Symbols Toggles symbol (ie letter) display at points on/off. (Default ***off***)

Lines Toggles lines between points on/off. (Default ***on***)

Labels Lets you redefine title and axis labels.

=> T/HIS Copies selected curves to linked T/HIS (when running) (see here)

Sel_All Selects all curves for plotting.

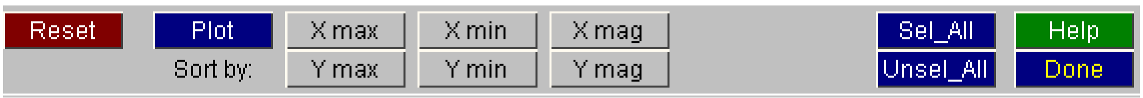
Unsel_All Deselects all curves for plotting.

Sort... Sorts the order of appearance of curves in the key

Filter... Filters visible curves by colour and symbol (FILTER Commands)

9.9.6.3. Sort... Sorting the Order of Appearance of Curves in the Key

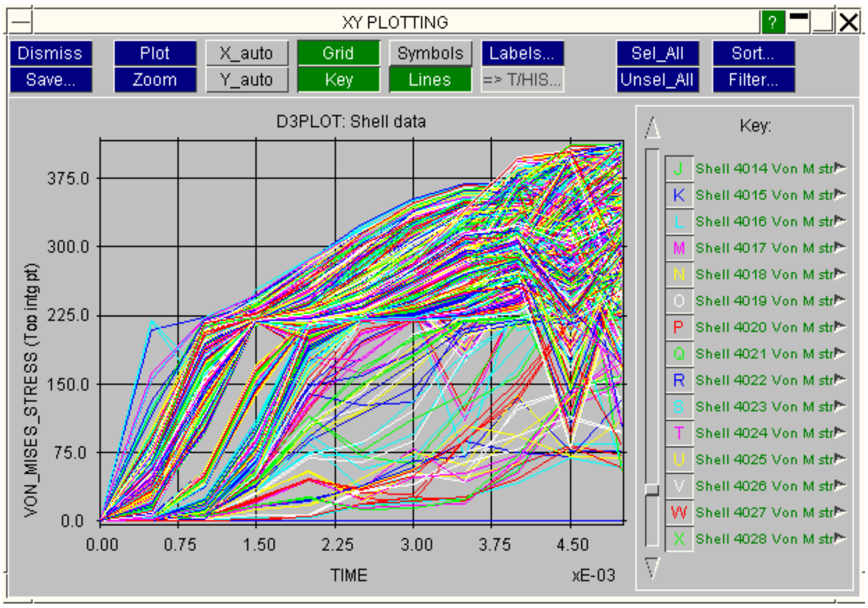
Sort... Sorting the order of appearance of curves in the key



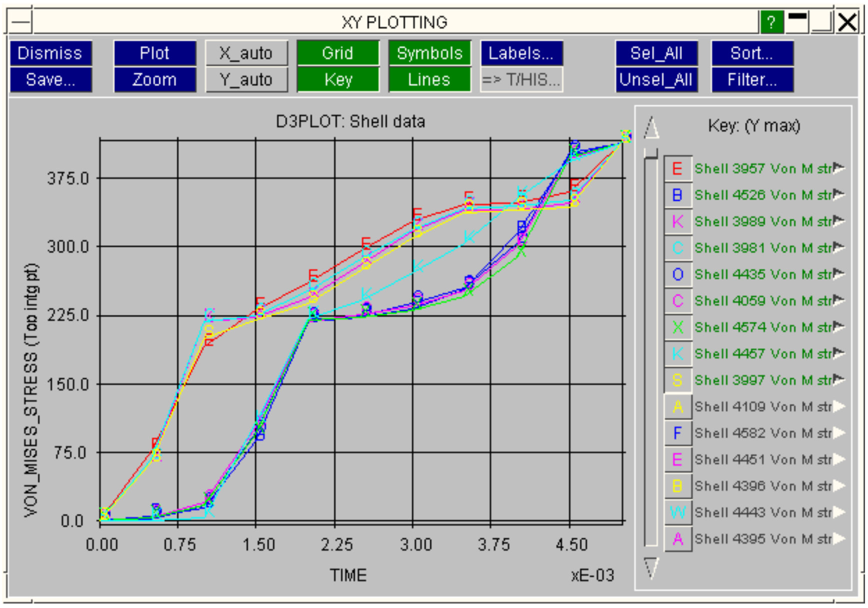
By default the curve "key" will appear in order of item definition, sorted by ascending label number. However when confronted by a plot with many curves often it is useful to be able to pick out the largest or smallest, and this can be achieved with **Sort...**.

You can sort by any of the criteria X max/min/mag, Y max/min/mag, and the effect will be to reorder the curve key as appropriate.

For example: here is a plot of von Mises Stress for 4624 shells over time - confusion - which is the highest value?

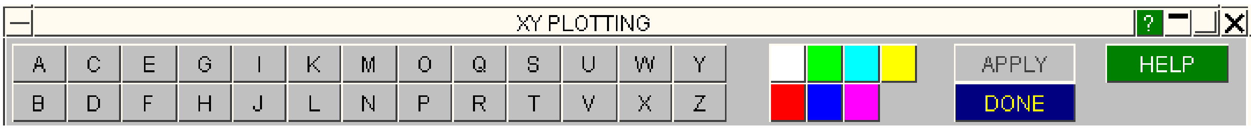


Using **Sort...** by Y max reorders the key with the top Y values at its top, making it possible to turn on just the top few curves, which are plotted below.



9.9.6.4. FILTER Commands

FILTER Commands



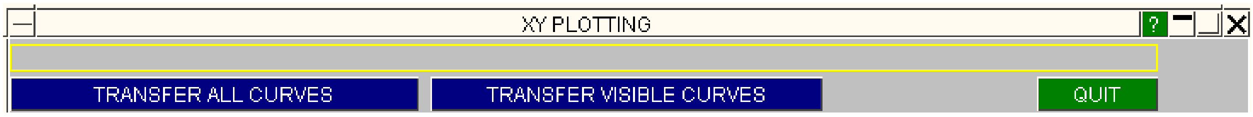
When you have many curves the **FILTER** option is another way of limiting what is displayed.

Clicking on a letter selects only the curves with that symbol, and on a colour only those of that colour. You can use both at once to restrict selection to just curves of that letter/colour combination. Use **APPLY** to plot just those selected, and **DONE** to return to the main plotting options menu.

9.9.6.5. => T/HIS Command: Export Curves to Linked T/HIS Session

=> T/HIS command: Export curves to linked T/HIS session

If the D3PLOT <=> T/HIS link has been invoked it is possible to export selected curves to the linked T/HIS session.



The number of free curves available in T/HIS is shown, and you can choose which of the local XY_DATA curves to export. These will become the next free #n curves in T/HIS.

(If linked T/HIS has not been invoked via the THDATA command the **=> T/HIS** button will be greyed out, and this function will be unavailable.)

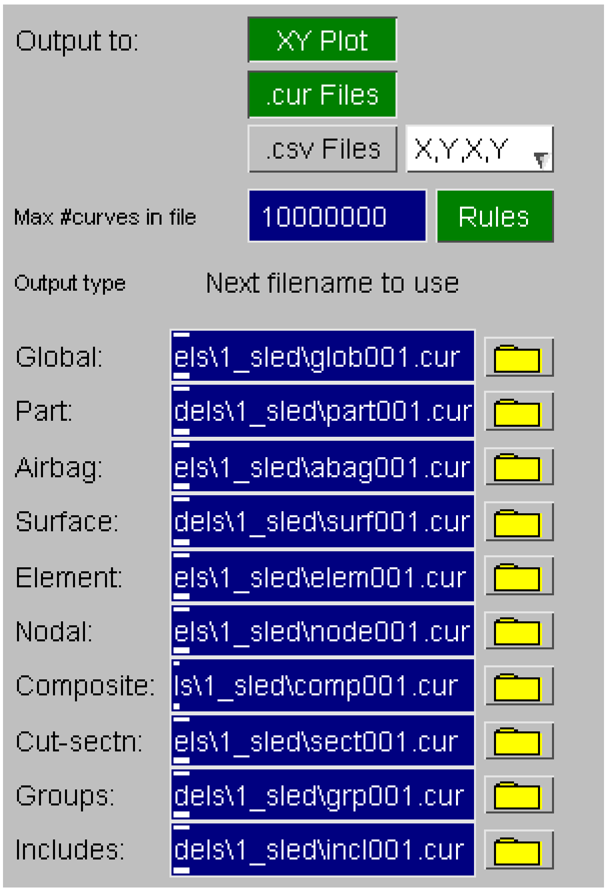
9.9.7. Managing "Curve" File Output

Managing "curve" file output

Output from **XY_DATA** can be written to "curve" or "csv" files as well as plotted.

This is controlled by the **.cur FILES** or **.csv FILES** commands on the main control panel: when turned off output to file is suppressed.

(Nb:You can also write files from the XY plotting window using the **SAVE...** command there.)



9.9.7.1. File Naming Conventions

File naming conventions

All "curve" and "csv" files **must** have a name in the form **<name>NNN.<extension>** , where:

<name>

is any sensible character string, but see conventions below;

NNN

is a 3 or 4 digit number using leading zeros if required;

<extension>

is any sensible character string. ".cur" and ".csv" is recommended.

The reason for this syntax is that more than one file may be generated by an output command, and some sort of systematic naming convention is required.

By default the following filenames are used:

For global model derived data vs. time:	`glob001.cur/csv`
For part derived data vs. time:	`part001.cur/csv`
For contact surface derived data vs. time:	`surf001.cur/csv`
For element derived data vs. time:	`elem001.cur/csv`
For nodally derived data vs. time:	`node001.cur/csv`
For any **COMPOSITE** data:	`comp001.cur/csv`
For cut-section data:	`sect001.cur/csv`
For group data:	`grp001.cur/csv`
For include file data:	`incl001.cur/csv`

You can change them if you wish, but it is recommended that you do not. If you omit the **NNN** part of the filename string D3PLOT will insert it for you.

9.9.7.2. Maximum Number of Files in a Sequence

Maximum number of files in a sequence

Implicit in the naming convention above is the sequence

<name>001.cur to <name>999.cur followed by <name>1000.cur to <name>9999.cur

Thus there is a maximum of 9999 files in a sequence.

Each new file output operation starts a new curve file, and D3PLOT will enforce the addition of integers to the base filename to ensure that each file has a unique name.

9.9.7.3. Defining the Maximum Number of Curves in a File

Defining the maximum number of curves in a file

From V92 onwards the default is for 10,000,000 curves to be written to a "curve" file. When the limit is reached the current file is closed, and the next one in the sequence is opened for output. In practice this means that all output from a single curve output command will be written to a single file. This limit does not apply to csv files.

You can change this value to any positive integer with the **Max #curves in file:** value.

9.9.7.4. Rules for File Output

Rules for file output

The following rules for file output are adopted:

(1)

Each output command **always** starts a new file.

(2) Existing files are **never** overwritten. Before a new file is opened a test is made to see if one of the same name already exists and, if it does, the **NNN** value is incremented until a vacant filename is found.

9.9.7.5. Format of a T/HIS "Curve" File

Format of a T/HIS "curve" file

This is an ASCII file which is organised as follows:

Line 1:	Graph title	(CHARACTER string)
Line 2:	X axis label	(CHARACTER string)
Line 3:	Y axis label	(CHARACTER string)
Line 4:	Curve label	(CHARACTER string)
Line 5:	<X value> <Y value>	(REAL numbers)
Line 6:	<X value> <Y value>	(REAL numbers)
:		
Line n:		(REAL numbers)

	<X value> <Y value>	

And if a second or subsequent curves are to be defined in the file:

Line n+1:	CONTINUE	(CHARACTER constant)
Line n+2:	Graph title	(CHARACTER string)
Line n+3:	X axis label	(CHARACTER string)
Line n+4:	Y axis label	(CHARACTER string)
Line n+5:	Curve label	(CHARACTER string)
Line n+6:	<X value> <Y value>	(REAL numbers)
:		

And so on for further curves.

Further notes on T/HIS curve file format:

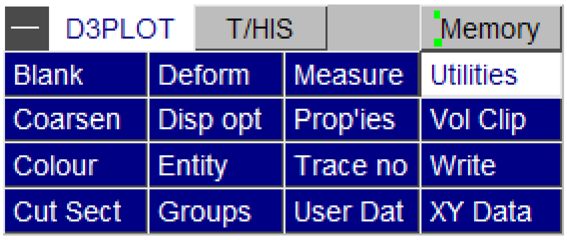
- Character strings should not exceed 48 characters.
- Numbers may be entered in free format, separated by spaces or commas. It is recommended that a field width of 20 be used (use a Fortran format statement something like 2e20.8) since this will then be compatible with Load Curve input in LS-DYNA.
- A curve may contain any number of <data> pairs.
- A file may contain any number of curves, separated by CONTINUE statements.

For more information see the T/HIS User's Manual.

9.10. UTILITIES Miscellaneous Utility Functions

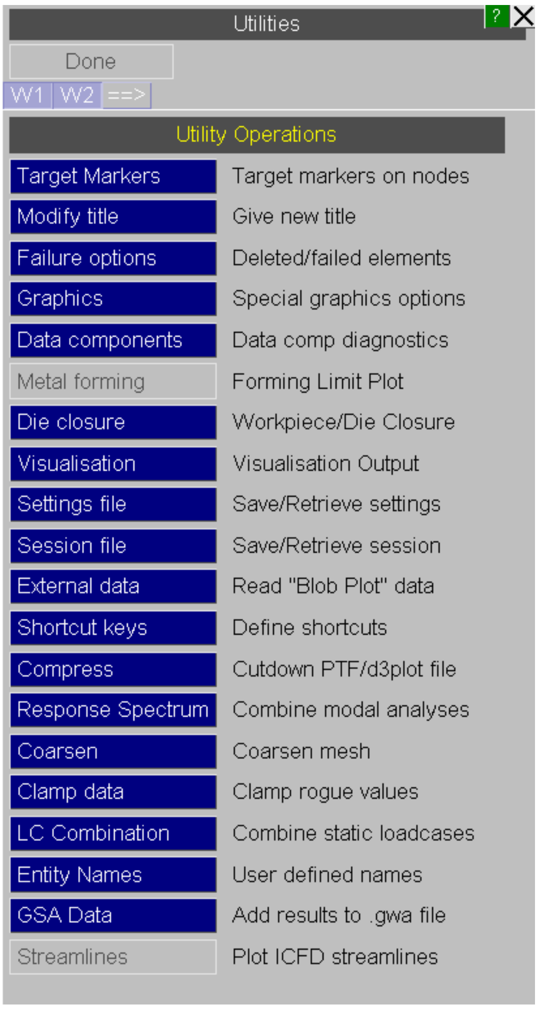
UTILITIES Miscellaneous utility functions

The UTILITIES menu is shown in the figure:



It provides a home for commands that have no other more logical place in the D3PLOT programme structure, and also for functions under development.

As the programme develops the latter commands will move out to their own menus.



9.10.1. TARGET_MARKERS Adding "Target" Symbols on Nodes

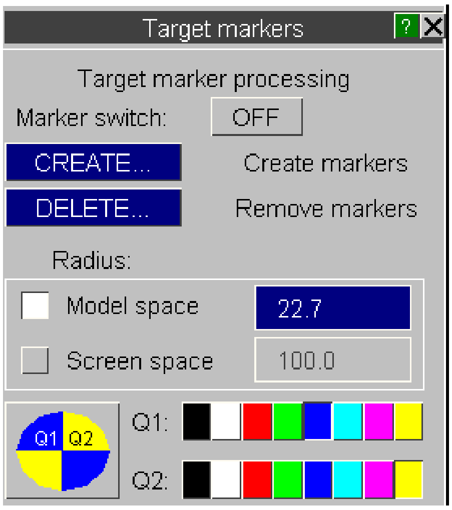
TARGET_MARKERS Adding target symbols on nodes

The figure (right) shows the **TARGET_MARKERS** control panel.

Target markers are used to mimic the actual markers that are attached to objects when being tested. This allows direct visual comparison between points on high-speed film and analytical results.

Because markers are attached to nodes they can only be defined in one model at a time. If multiple models are present you will be forced to choose the model in which to define the markers.

Markers are stored as a "per model" attribute, so if you have multiple windows in a model the markers will appear in all of its windows.

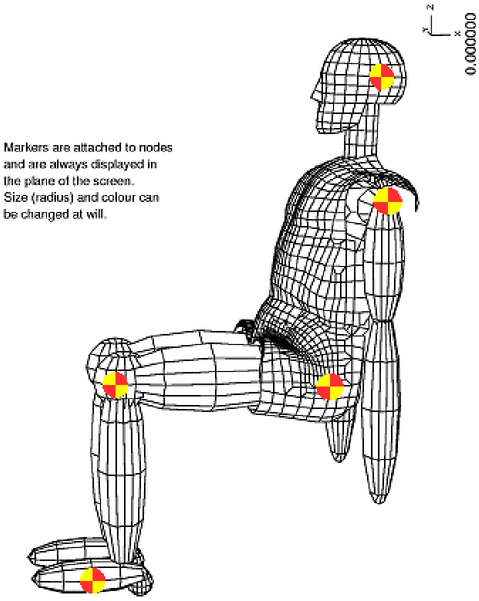


The figure (right) shows an example of a dummy model with target markers applied.

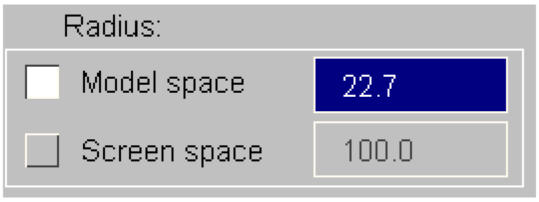
Markers are made up a circle with quadrants of alternating colours.

They are attached to nodes and are drawn only if their parent node is visible.

They are always drawn in the current plane of the screen regardless of the true orientation of any underlying mesh.



Marker Radius



If expressed in model space, the default, marker radius will be about 3% of the model overall dimension, and size will change as you zoom in and out.

If expressed in screen space marker radius will be a fixed size, independent of the model dimensions, and will not scale as you zoom in or out.

Marker switch: Controlling display of target markers

By default this switch is on, and any markers that are defined will be drawn when the plot is next updated. To suppress the display of markers without having to delete their definitions turn this switch off.

CREATE Creating target markers at nodes

This is very simple. Define a <list> of nodes using the standard range definition panels, and the next time the model is drawn (with the Marker switch on) target markers with the current attributes will be drawn on those nodes.

DELETE Removing target markers from nodes

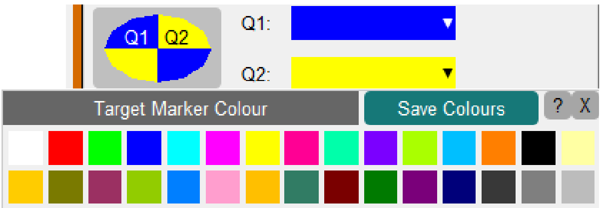
This is the exact opposite of **CREATE** above: define the <list> of nodes from which target markers are to be removed. It doesn't matter if a node is selected that does not currently have a marker attached.

Radius Defining the symbol radius

A default radius for markers (in model space units) is computed based on the model's longest dimension: you can alter this at will. This takes effect the next time you update the plot.

Setting quadrant colours

The default colours for the marker symbol are yellow and blue, but you can choose new colours for the two quadrants (**Q1** and **Q2**) from the standard palette given. These take effect the next time you update the plot.



Notes on target markers

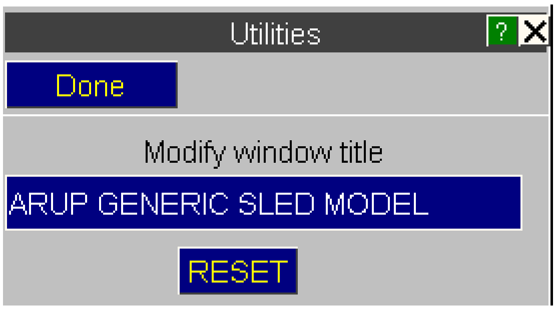
Note 1: Target markers are always drawn in the plane of the screen, regardless of model orientation. On 3-D devices this results in the symbols being rotated as the image is rotated, and they can disappear when edge-on after a 90 degree rotation. To see them again you will have to issue an explicit redrawing command in the new orientation.

Note 2: Hidden surface removal is based on the coordinates of the parent node only: markers are drawn in full if the node is visible, otherwise not at all.

9.10.2. MODIFY_TITLE... Changing the Title String Used for the Header on Plots

MODIFY_TITLE... Changing the title string used for the header on plots

The analysis title taken from the `.PTF` file is normally used for the header on plots, and also for headers in output files.

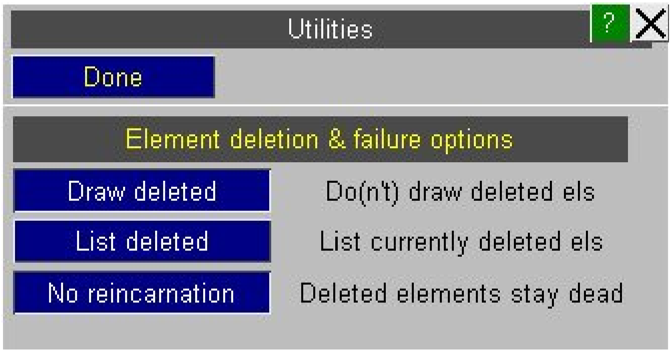


You can change this by simply typing in a new string, followed by **DONE** . This will take effect the next time you update the plot.

Changes here have no effect on the title stored permanently in the `PTF` file.

9.10.3. FAILURE_OPTIONS... Controlling and Listing Deleted and Failed Elements

FAILURE_OPTIONS... Controlling and listing deleted and failed elements



This figure shows the **FAILURE_OPTIONS** control panel.

This permits you to control the display of, and to list, elements flagged as "deleted" by LS-DYNA.

A "deleted" element is one that is flagged in the database file by LS-DYNA as having been deleted in the course of the analysis.

9.10.3.1. DRAW_DELETED Switching the Display of Deleted Elements On/Off

DRAW_DELETED Switching the display of deleted elements on/off

By default deleted elements are not drawn. Once they have been flagged in a plot state as deleted they are removed from the display list altogether. If you turn this switch on they will be drawn.

WARNING: Displaying deleted elements can occasionally crash D3PLOT. Nodes on deleted elements may become unrestrained and, if subject to external force, accelerate off in an arbitrary direction at impossibly high speeds.

Attempting to display facets attached to such nodes, which may have one corner in your model but another somewhere in Ursa Minor, can cause problems for the hidden-line algorithms.

HINT: A better way to visualise deleted elements is to leave this switch ***off*** , and instead to use the **DT_DELETION_TIME** component in a solid contoured plot (ie **CT** or **SI** modes). This draws the model with undeformed coordinates, but with deleted elements at the current time shown in a different colour.

9.10.3.2. LIST_DELETED Listing Deleted Elements on the Display

LIST_DELETED Listing deleted elements on the display

For each element type category a list of elements deleted at the current time is given in a listing box on the display.

9.10.3.3. No_Reincarnation Stopping Deleted Elements "Coming Back to Life"

No_Reincarnation Stopping deleted elements "coming back to life"

LS-DYNA writes deletion tables at the end of every state in the output file which set a flag against any element that has been deleted.

In normal usage elements start off "alive", then "die" (get deleted) if their material model implements failure and they are deemed to have failed, and they remain dead for the rest of the analysis. However some keywords in LS-DYNA use these tables in a slightly different way, for example `*DEFINE_CONSTRUCTION_STAGE` , can result in an element starting off "dead" and then coming to life at some later stage in the analysis. Therefore D3PLOT cannot adopt the "once you are dead you remain dead" approach as its default, and instead the deletion status of every element at every state is stored so that elements can die and come back to life at will.

However there is a bug in some versions of LS-DYNA where elements that "die" due to material failure are (correctly) marked as deleted at that state, but somehow this information is lost later on in the analysis and they appear to come back to life again. Since deleted elements have no stiffness they can develop huge deflections, and drawing these can mess up plots horribly, so a means of solving this problem is required, and the **No_Reincarnation** feature provides this.

Clicking on **No_Reincarnation** does the following:

- For every model in the database D3PLOT starts at state 1 and works its way linearly forwards to the last state, propagating the deletion status of elements forward by setting the deletion status of an element in state `<i>` to be a logical OR of that and its status in state `<i-1>` . Therefore once an element is deleted it cannot come back to life again.

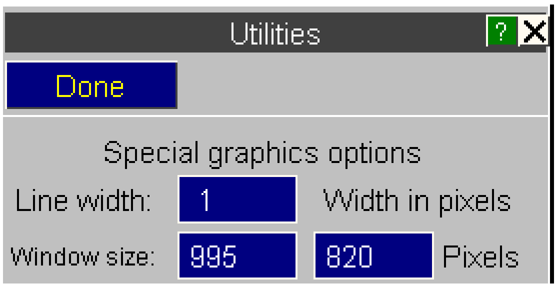
- This process may be slow if the analysis has many states since D3PLOT has to read the deletion status of each state if it is not already in memory. States will only be in memory if they have been displayed, so if you have read in a model and then jumped directly to the last state it will have to read the deletion tables of all the intervening states.
- If set this flag remains in force for any model read subsequently.
- If unset reading of any subsequent models will behave normally, ie deletion status in each state will revert to being independent of any other state. However propagation of deletion in models to which this process has already been applied is not undone - once applied to a model this process is irreversible, and if you need to revert to normal deletion behaviour in that model you will have to **Reread** it.

9.10.4. GRAPHICS... Setting Special Graphics Parameters

GRAPHICS... Setting special graphics parameters

This figure shows the **GRAPHICS** control panel.

This lets you set special attributes of the graphics display.



9.10.4.1. Line Width: Setting the Line Width in Pixels

Line width: Setting the line width in pixels

By default all graphics lines are drawn 1 pixel wide, but on hardware that supports it you can increase this to <N> pixels.

A common reason for doing this is to improve the appearance of images being transferred to avi or mp4. Lines taken from a high resolution computer graphics display often become patchy in movie files, so increasing the line thickness greatly improves their visibility.

You will notice that drawing takes much longer with thick lines: this is because the graphics device has to render more pixels.

9.10.4.2. Window Size: Setting the Graphics Window Size in Pixels

Window size: Setting the graphics window size in pixels

On windows displays it is sometimes convenient to be able to set the graphics window to a specific size in pixels. This command allows you to do that: in screen menu mode the graphics window only is resized, in raw windows mode the whole display window is resized.

The window size is **not** "clamped" by this command: subsequently resizing the window will overwrite this setting.

If you have multiple windows the size will be set for all windows with **`W1 .. Wn`** tabs active.

9.10.5. DATA_COMPONENTS Listing the Current Database Contents to Screen and Current Data Components to File

DATA_COMPONENTS Listing the current database contents to screen, and current data components to file

This command is an aid to debugging the software, so you don't have to know how to use it!

It lists to the screen and to file **`diag.out`** the following table:

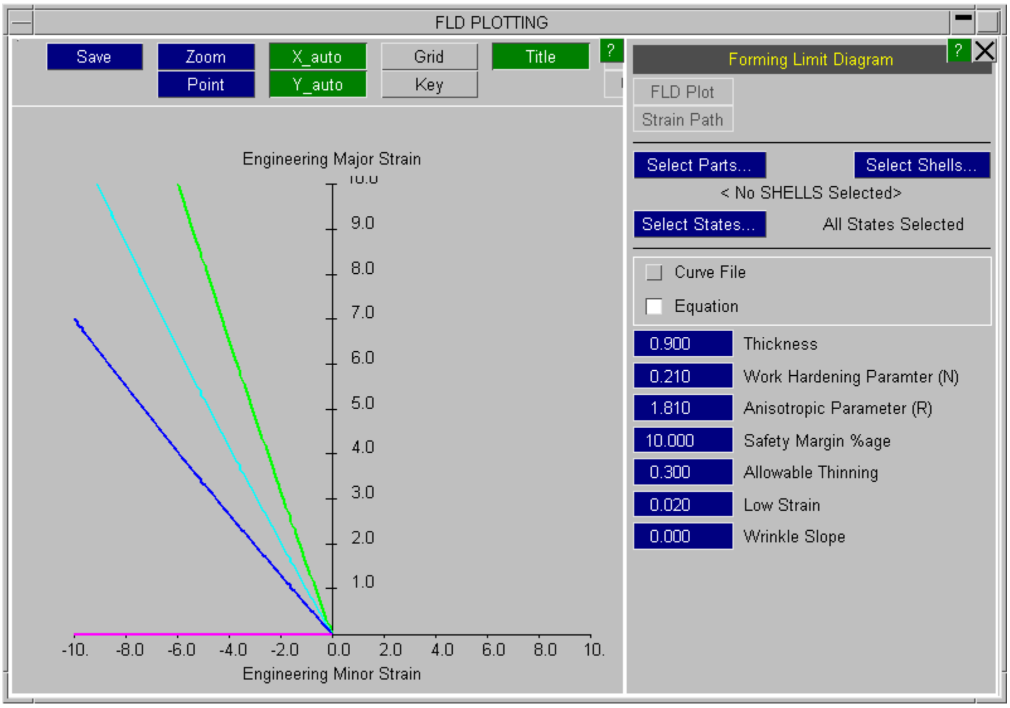
```
D3PLOT version 7.0: Data component summary printout

=======================================
# in database          Solids       Thin shells   Thick shells    Beams
                       ( 16)        ( 4624)       ( 0)            ( 0)
Stress tensor          6 * 1        6 * 3         6 * 3           n/a
Plastic strain         1 * 1        1 * 3         1 * 3           n/a
Strain tensor          6 * 1        6 * 2         6 * 2           n/a
Extra variables        0 * 1        0 * 3         0 * 3           n/a
Forces & moments       n/a          8             n/a             6
Thickness & energy     n/a          4             n/a             n/a
Plastic beam data      n/a          n/a           n/a             0
Deletion flagged       T            T             T               T
```

In addition it writes to file only a listing of the D3PLOT internal data component tables. These are bitwise encoded listings cross-referencing data components against element type and other internal flags. They have no significance to users.

9.10.6. METAL FORMING

METAL FORMING

The **METAL FORMING** panel contains a number of special options and plotting modes for use with metal forming analysis. The figure (below) shows the basic **METAL FORMING REFERENCE** control panel.



This panel consist of three sections.

1. The top part of the panel accesses a number of special plotting modes, see METAL FORMING plotting options .
2. The middle part of the panel allows entities to be selected for some of the plotting options in the top part of the panel and the states to use for the Strain Path calculation (all by default), see METAL FORMING Entity and State Selection .
3. The bottom part of the panel allows a number of variables to be entered which are used by some of the plotting options in the top part of the panel, see METAL FORMING Data Values

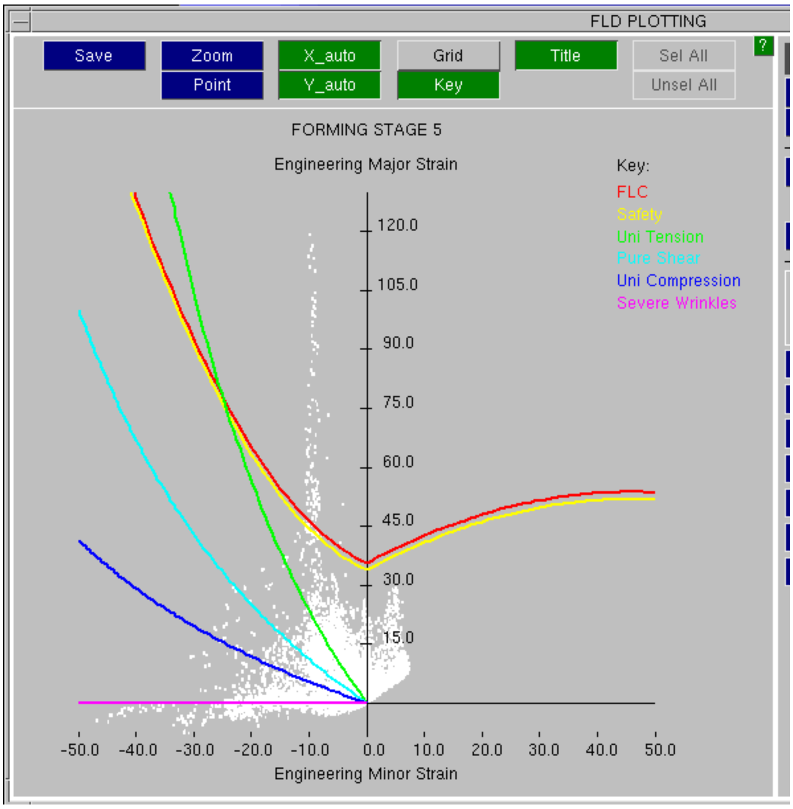
9.10.6.1. METAL FORMING Plotting Options

METAL FORMING plotting options

The top part of the **Metal Forming Panel** contains two special plotting options.

FLD PLOT - Forming Limit Diagram

This option will only be available if some entities have been selected for plotting, see METAL FORMING entity and state selection . When selected, this option will produce a graph of **Major Engineering Strain** v **Minor Engineering Strain** at the current analysis time for the selected entities, see the figure below.



In addition to the shell elements that are represented by white dots, the graph also contains a number of lines relating to physical and artificial phenomena.

FLC	Forming Limit Curve. This curve represents the point at which the material would fail. This curve may be generated from an equation or read in from a file, see METAL FORMING Data Values for more details.
Safety	This curve represents a safety margin. It is generated automatically by subtracting a constant strain percentage from the FLC, see METAL FORMING Data Values
Uniaxial Tension	The uniaxial tension curve is defined by

$$\mathbf{E}_{(major)} = \mathbf{E}_{(minor)} \times -\left(\frac{1+R}{R}\right)$$

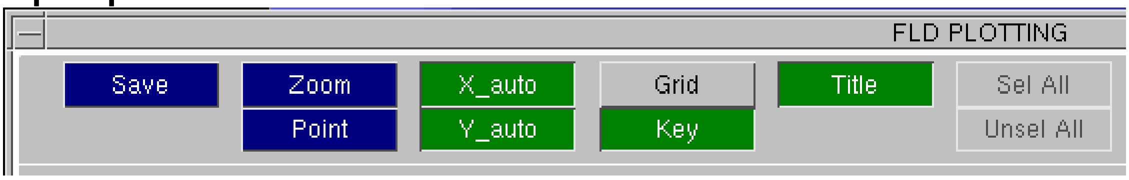
Pure Shear	The pure shear curve is defined by

$$\mathbf{E}_{(major)} = -\mathbf{E}_{(minor)}$$

Uniaxial Compression	The uniaxial compression curve is defined by $$E_{(major)} = E_{(minor)} \times -\left(\frac{R}{1+R}\right)$$
Severe Wrinkles	The severe wrinkles curve is defined by **E** ***(major)*** **= -** Wrinkle slope **E** ***(minor)***

where	**E** ***(major)*****, E** ***(minor)***	True major/minor strains
	R	Normal Anisotropy Parameter

Graph Options



Save	Save the points and the curve data to a curve file for reading into T/HIS.
Zoom	Uses the cursor to zoom in on a rectangular area of the graph, and enlarges the area to fill the whole window.
Point	Select a point on the graph. The ID of the nearest shell to the point is then written out to the dialogue area and the shell is labelled in the graphics window if it is visible.
X_auto	Resets the X scaling to automatic.
Y_auto	Resets the Y scaling to automatic.
Grid	Toggles a grid at the tick mark intervals on/off (Default ***off***).
Key	Toggles the curve "Key" on/off (Default ***off***).
Title	Toggles the analysis title on/off (Default ***on***).
Sel All	Blanks all Strain Path curves (for strain path plot).
Unsel All	Unlanks all Strain Path curves (for strain path plot).

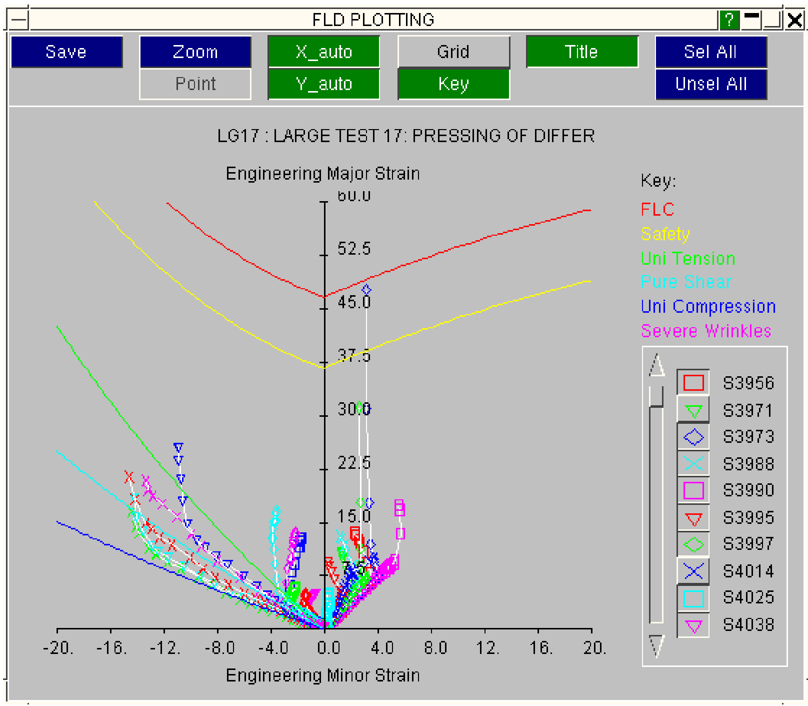
STRAIN PATH

The **STRAIN PATH** option allows the user to plot the **Major Engineering Strain** v **Minor Engineering Strain** path for the selected entities, see METAL FORMING Entity and State Selection . This produces a plot of **Major Engineering Strain** v **Minor**

Engineering Strain at every plot-state, see the following figure. This option is useful as it shows the strain-route taken by an element. For example, if the material properties defined by the user do not allow failure then this plot may help to highlight elements that appear to be okay at the end of the analysis but are likely to have failed during the analysis if failure criteria had been included in the material model.

If the selected entities are shells then D3PLOT will display the first 20 shells that have been selected. If a material has been selected then D3PLOT will only display the first 20 shells in that material.

Curves can be blanked and unblanked using the key or **Sel All** and **Unsel All** options.



FORMABILITY Plots

In D3PLOT 13.0 onwards, select Data Component **FP_FORMABILITY** to plot the formability of the product.

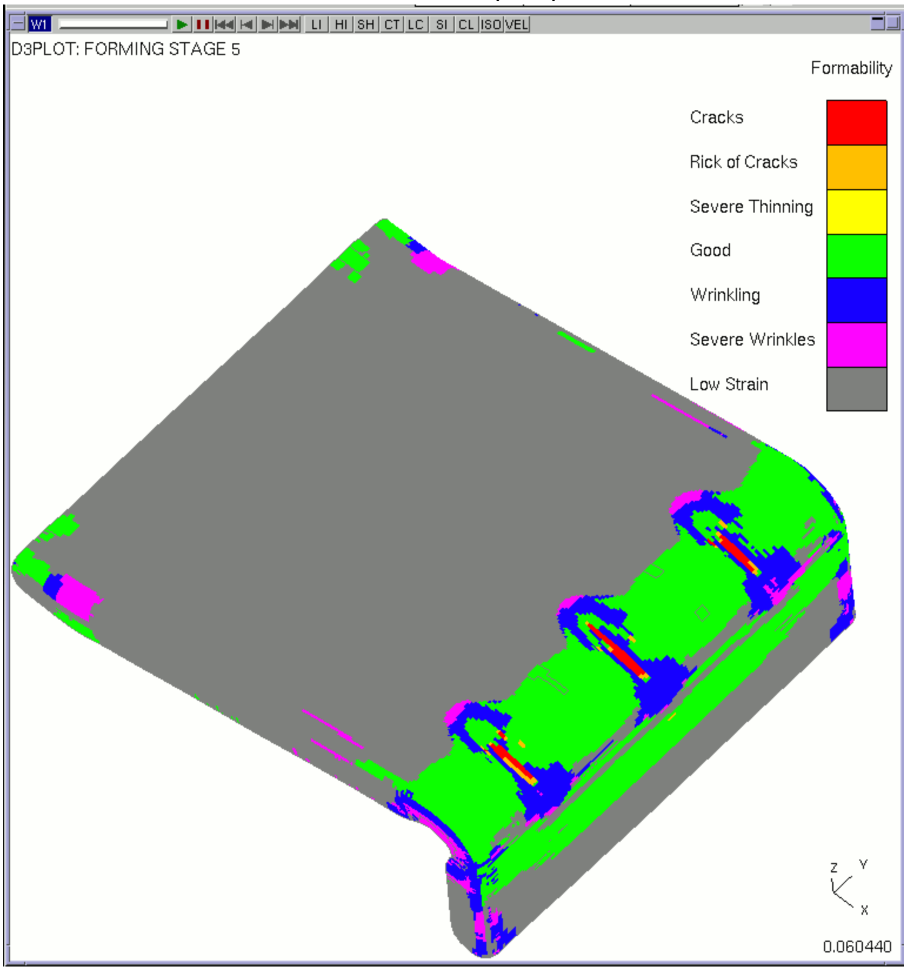
The formability of each element is determined by its position on the FLD as follows:

Cracks	Above the FLC.
Risk of Cracks	Between the FLC and the safety curve.
Severe Thinning	$1 - 1/\exp(E_{(major)} + E_{(minor)}) \leq$ Allowable Thinning

Good	Between the safety curve and the uniaxial tension curve, not in the region of severe thinning.
Wrinkling	Between the uniaxial tension curve and severe wrinkles curve.
Severe Wrinkles	Below the severe wrinkles curve. (For default wrinkle slope 0.0 this corresponds to any value with -ve major strain.)
Low Strain	$\sqrt{(E^2_{(major)} + E^2_{(minor)})} \leq \textit{Low Strain Value}$
where	$E_{(major)}$, $E_{(minor)}$ Engineering major/minor strains

Before generating a **FORMABILITY** plot, the option to have 8 contour levels is automatically selected (see DATA COMPONENTS - BASIC). After the plot is completed these values are reset to their previous settings.

FORMABILITY (**CT**) Plot

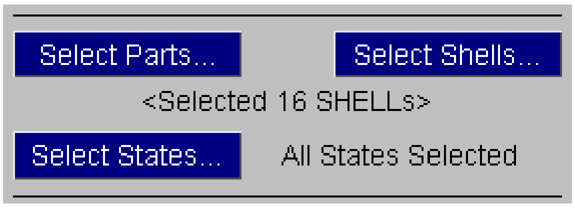


9.10.6.2. METAL FORMING Entity and State Selection

METAL FORMING entity and state selection

The metal forming options only apply to 3 and 4 noded shell elements and therefore only shells or shell materials can be selected. The panel also shows how many shells have been selected.

The Strain Path calculation plots data for the states selected here. By default all states are selected.



9.10.6.3. METAL FORMING Data Values

METAL FORMING data values

The **METAL FORMING** panel allows a Forming Limit Curve (FLC) to be read in from a curve file (see figure right), or to be generated by D3PLOT from an equation (see figure below right). The curve file name is entered in the box.

The following parameters are added to generate the other curves:

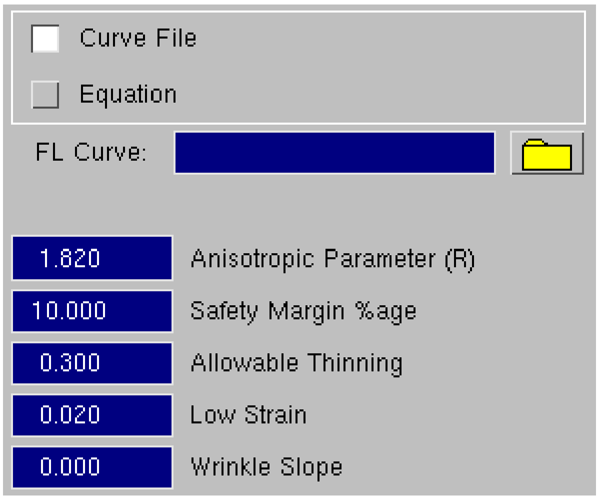
Anisotropic Parameter (R)

Safety Margin %age

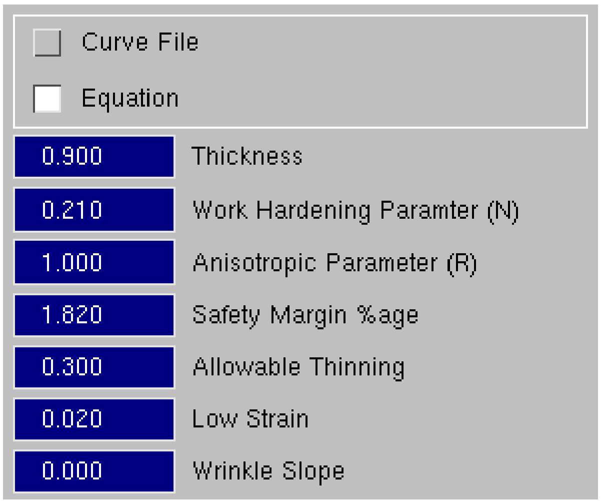
Allowable Thinning

Low Strain

Wrinkle Slope



If the FLC is generated from an equation the following parameters are entered in the boxes: **Thickness** , and **Work Hardening Parameter (N)** .



The FLD curves are then generated according to an empirical formula which is generally applicable for steels as follows:

If E[minor] < 0	**E[major] = FLD + E[minor] * (0.042 * E[minor]+0.627)**
If E[minor] > 0	**E[major] = FLD + E[minor] * (-0.0086 * E[minor]+0.785)**

Where	**FLD**	=	**(N/0.2116) * (23.36 + 14.042*min(T, 3.0))**
	N	=	**Work Hardening Parameter**
	T	=	**Material Thickness (mm)**

Other data values

Anisotropic Parameter (R)	Used to automatically generate uniaxial tension and uniaxial compression curves, see FLD PLOT - Forming Limit Diagram .
Safety Margin %age	The safety curve on the **FLD** and **STRAIN PATH** plots is automatically generated by subtracting the constant percentage of strain **Safety Margin %age** from the FLC.
Allowable Thinning	Severe thinning occurs if $1 - 1/\exp(E_{(\mathrm{major})} + E_{(\mathrm{minor})}) \leq \text{Allowable Thinning}$
Low Strain	Strains are ignored if $\sqrt{(E^2_{(major)} + E^2_{(minor)})} \leq \textit{Low Strain Value}$
Wrinkle Slope	Sets the gradient of the severe wrinkles curve, see FLD PLOT - Forming Limit Diagram .
where	**E** ***(major)*** **,** **E** ***(minor)*** Engineering major/minor strains

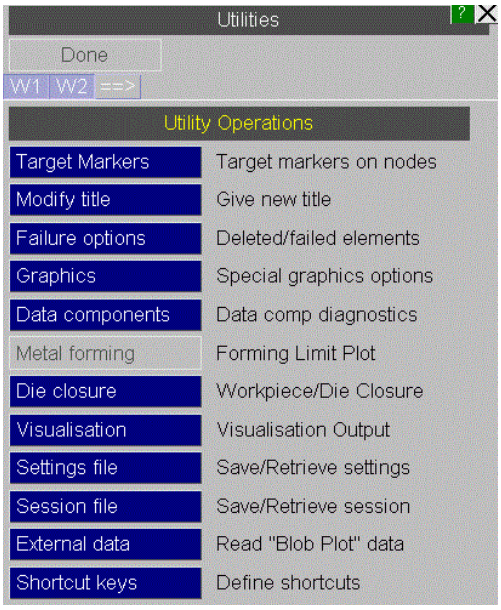
9.10.7. UTILITIES, DIE_CLOSURE

UTILITIES, DIE_CLOSURE

The "Die-Closure" command set, found in the **UTILITIES** menu creates and controls the special **CLOSURE** data component.

This data component calculates the distance remaining before a "workpiece" makes contact with its "die".

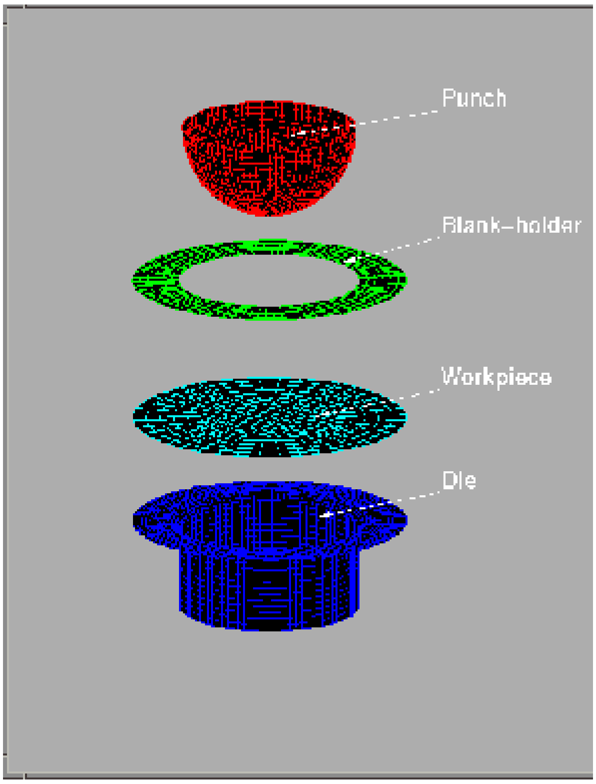
The terminology implies metal-forming, and the most common use of this will be to determine how much further a deformable blank has to be pressed in order to achieve its desired shape. But any problem requiring the distance between two pieces of 2D/3D structure to be calculated can be solved here, and it has application to many cases in which "distance to contact" needs to be known.



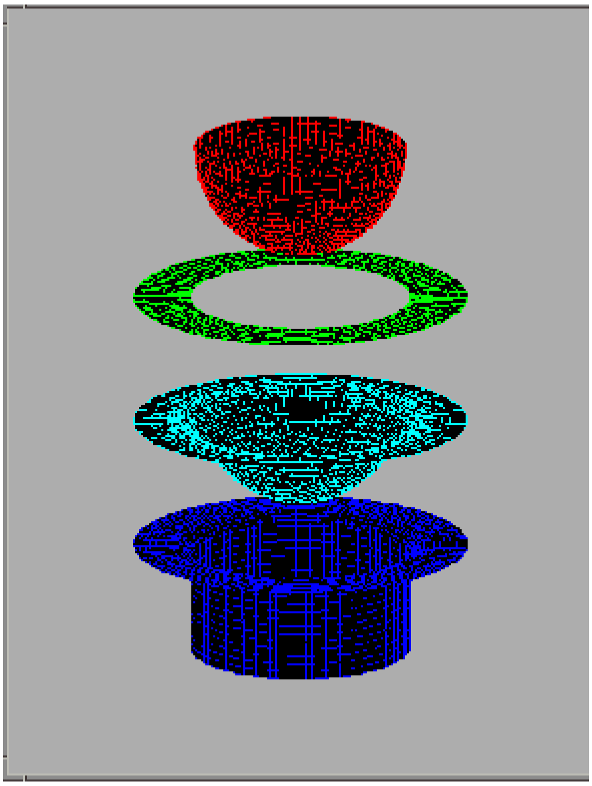
9.10.7.1. Defining the Problem

Defining the problem

The two figures below show a simple pressing problem, parts exploded, in which the workpiece (blue) is pressed by a punch into a cup shape. We need to know how far the punch (red) must be pushed down to achieve the required final shape.

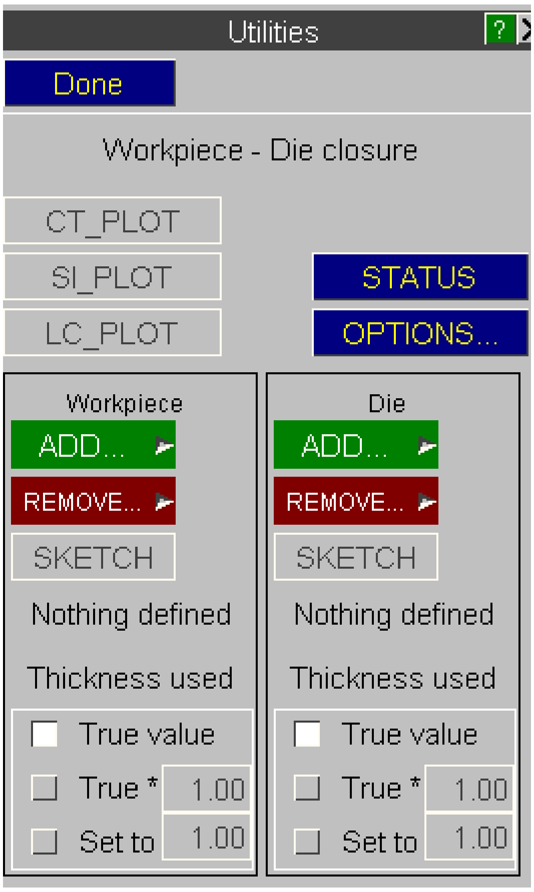


Intial problem layout


Final shape

9.10.7.2. Defining the "Workpiece" and "Die" Sides

Defining the "Workpiece" and "Die" sides



Closure distance is computed between nodes on the "workpiece" and element faces (facets) on the "die". Therefore it is necessary to tell D3PLOT what each side actually is, as in LS-DYNA contacts.

Once a type has been chosen the standard D3PLOT selection menu is used to define the actual elements or parts.

Each side may be any mixture of 2D and/or 3D elements: solids, shells, and thick shells, although in most cases (as here) shells will be used. Sides may be defined by element and/or part, and may have elements added and removed at will, and elements may be deformable or rigid. Definition by part has advantages in adaptively remeshed cases, as in this example, since the part definition is "remembered" across remeshes and the extra elements are auto-matically included in the relevant side.

Elements may not exist on both sides at once, and adding an element to one side that already exists in the other will cause it to be removed from the older definition. This cross-check is automatic. The definition panel is updated immediately to show how many elements there are on each side, and you can **SKETCH** each side at any time to confirm your choice.

9.10.7.3. Plotting Closure

Plotting closure

Once there is at least one element on each side the **CT_PLOT** , **SI_PLOT** and **LC_PLOT** buttons will be made "live", and these will immediately compute the closure values, set the programme-wide data component to **CLOSURE** , and perform a plot in the relevant mode (continuous-tone, shaded image, line contours).

You can also select the **CLOSURE** data component explicitly from the **DISPLACEMENTS...** data category, and plot it using main menu commands. It is also a scalar (nodal) data component that can be used in **WRITE** and **XY_PLOT** contexts.

9.10.7.4. How the "Closure" Calculation Works, and the Output it Produces

How the closure calculation works, and the output it produces

It is necessary to calculate the distance from each workpiece node to the nearest facet on the die. This is done by determining which facet a node will be projected onto when contact is made, and computing to the vector distance (less element thicknesses) between node and facet. This is very similar to the `*CONTACT_NODES_TO_SURFACE` problem in LS-DYNA, except that D3PLOT must consider nodes at some distance from, as well as in contact with, the die surface.

For nodes close to or in contact with a facet this is a simple calculation, but when a node is quite a long way from the die problems of ambiguity ("Which facet will I ultimately make contact with?") and complexity ("I need to be tested against many distant facets") arise.

To get round these problems, and to achieve a reasonably short computation time, a bucket-sorting process is used:

- The volume of space around the workpiece, plus a margin, is calculated and then sub-divided by (x,y,z) coordinates into a 3D matrix of "buckets". Each bucket is a smaller, rectilinear volume of space.

- Each workpiece node is assigned to a bucket, based on its coordinate. Die facets may lie in one or more buckets, or none at all if very distant.
- For each die facet closure is only calculated for workpiece nodes that exist in the same or immediately adjacent bucket(s) as the facet.

D3PLOT uses a 10x10x10 (x,y,z) array of buckets, and chooses default dimensions such that closure values up to about 20% of the longest workpiece dimension will be calculated. Thus node closures will fall into one of four categories:

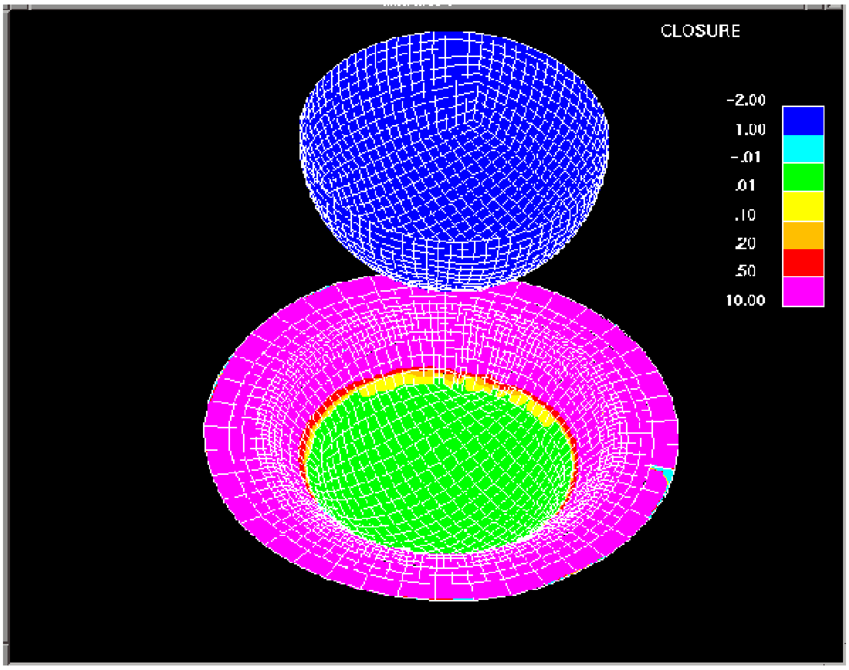
Node location	**Meaning**	**Value**
In Contact	The workpiece node lies on or behind the die facet surface. (The penalty algorithm in LS-DYNA means that a node may penetrate an element.)	0.0
Near to contact	The node is reasonably close to a facet, and a closure distance can be calculated.	> 0.0
Uncomputed	The node is too far from the die for a closure distance to be computed, or has fallen into a "tunnel" between facets (see below).	-1.0 [1]
Uninvolved	For nodes not in the workpiece, so not involved in the closure calculation at all.	-2.0 [1]

Closure categories and associated values

Note 1: These values are defaults which can be changed, see **OPTIONS...** below.

9.10.7.5. Typical Closure Output

Typical closure output

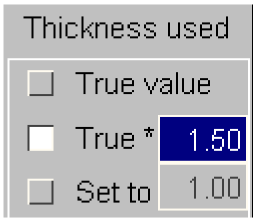


This figure illustrates the final state of the example shown earlier. Here the "die" side is actually the punch, since that is the shape against which we wish to measure closure.

Note that the contour levels have been chosen explicitly, with unequal intervals, to illustrate the following:

- The centre of the workpiece is fully in contact: green area, value = 0.0.
- There is a zone of progressive opening roughly half way up the walls, where a gap between workpiece and punch is opening up: yellow (0.01) to red (0.5).
- The rest of the workpiece has a gap of 10mm or more (purple).
- All punch (ie the "die" side) nodes, for which no closure values are calculated, are blue, having the value of -2.0.
- There is a small patch of "uncomputed" nodes (light blue, = -1.0) at about 4 o'clock on the workpiece. The reasons for this are explained later.

Controlling the die-closure calculation process



Controlling the thickness used

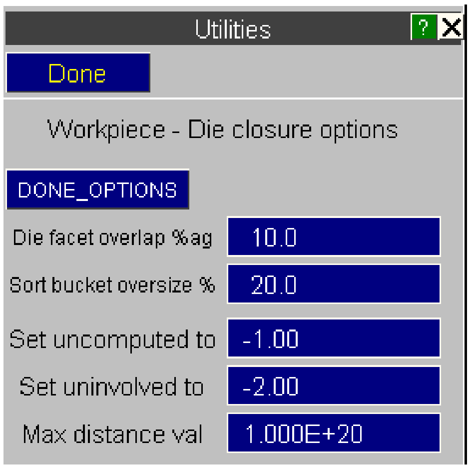
By default the true thickness of both workpiece and die is used when calculating closure. "True" thickness depends on the element type:

(Thin) shells	The current element thickness as reported in the `.ptf` file. The centreline +/- half this value is used.
Solids and Tk. Shells	Zero. (The nodes are assumed to lie on the face surface.)

For workpiece nodes the average thickness of all elements meeting at the node is used, for die facets the actual value is used.

For each side individually you can apply a factor to the "true" value, or override it with an explicit value. (You may wish to do this if you have used artificial thicknesses during the calculation, since this will affect contact geometry.)

OPTIONS... Controlling calculation parameters



The default parameters chosen by D3PLOT should be satisfactory in most cases, but there are situations when you may wish to alter them.

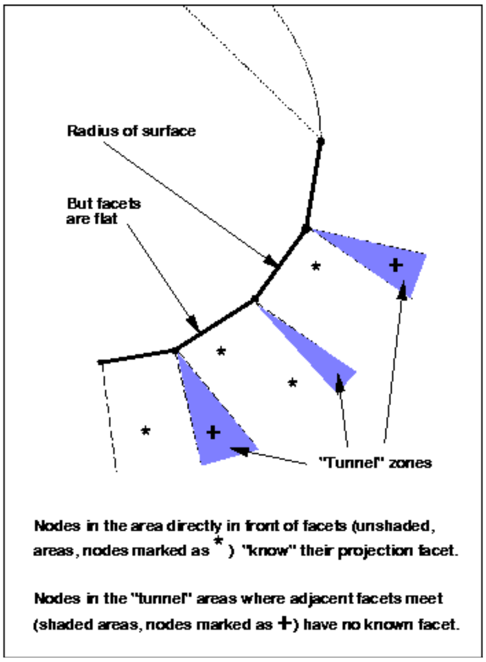
The **OPTIONS...** command gives access to the following adjustable parameters.

"Die facet overlap %age" controls the extent to which die facets are artificially enlarged (in the in-plane direction) for the purposes of checking workpiece node projection.

This is important where the die surface is convex with respect to the workpiece, since it helps to prevent workpiece nodes falling into the "tunnels" between projected facet volumes.

This can be illustrated as follows:

Using the Die facet overlap %age value to prevent nodes being lost on convex surfaces



When workpiece nodes are being projected onto a convex die surface problems can arise when nodes lie in the dead zones ("tunnels") where facets meet.

This is illustrated here: nodes marked * lie in the areas opposite die facets, and so "know" which facet they are likely to be projected onto.

Nodes marked + in the shaded tunnel areas don't "know" which facet they will be projected onto, and so no closure value is calculated for them.

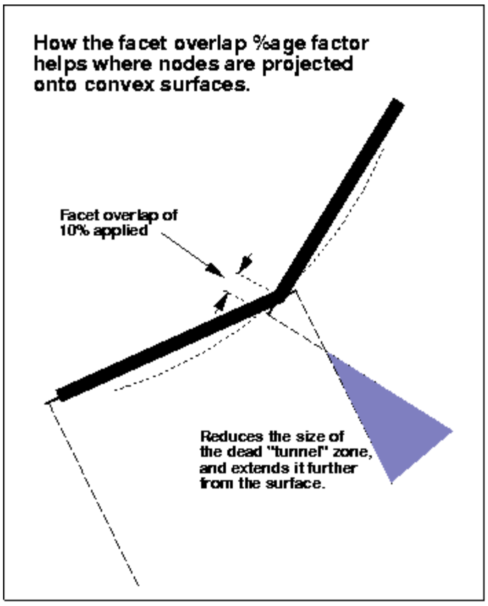
Clearly this problem becomes more acute as the distance of a workpiece node from the die increases, and is the main reason why closure values for such distant nodes may appear randomly to be classified as "uncomputed", and given a value of -1.0.

The situation can be improved by artificially increasing the width of die facets so that they overlap, using the "Die facet overlap %age" factor.

This is illustrated here: a value of 10% (the default) has been used, and this shows how the "tunnel" dead zone has been made smaller, and also extended further away from the die surface.

Clearly larger values will lead to fewer nodes being lost in "tunnels", but this must be balanced against the fact that it may lead to invalid closure values being calculated when nodes are projected onto the wrong facets because these have become artifically extended.

This is a case for engineering judgement.



Using the Sort Bucket oversize %age to include distant die facets

The volume of space used for bucket sorting is based on the dimensions of the workpiece, not the die. Therefore if the die is much larger, or some considerable distance away, then there is a good chance that many facets on it will lie outside the bucket volume, and this means that they won't be considered when computing closure.

Increasing the Sort bucket oversize %age value from its default of 20% may include these ...

BUT:

- Closure values calculated from distant facets are likely to be unreliable, since by the time the workpiece and die meet at that point their respective structures will probably have been deformed so much that contact will actually occur at some other location.
- Increasing the bucket volume will increase the number of nodes and facets in each bucket. The time taken to compute closure rises as a function of ($N_{Workpiece}$ * N_{Facet}) in each bucket, and these values increase as a function of bucket volume, ie linear dimension cubed. So increasing the bucket size can lead to a rapid rise in the time taken to compute closure, in the worst case by a fourth power.

Or, put another way, you need good reasons to increase this value!

Controlling the values used for uncomputed, uninvolved and distant nodes

These values do not affect the calculation at all, only the values assigned to special cases.

Set uncomputed to:	Is the value assigned to nodes on the workpiece for which no closure value can be computed. By default this is -1.0, but you can choose any value - although you should avoid values that might be confused with valid results (ie zero or small +ve numbers).
Set uninvolved to:	This is the value assigned to all nodes in the model that are not part of the workpiece. The default is -2.0, but again you can choose any sensible value.
Max distance val:	Nodes with computed closure values greater than this value have them reset to the "uncomputed" value. By default this is +1.0e20, ie no nodes will ever fall into this category, but you can set it to a sensible upper-bound value.

The advantage of using negative values for uncomputed and uninvolved nodes is that they will never be valid closure distances, and so can be isolated during contouring by judicious choice of contour bounds, or excluded from plots altogether with the **Limiting Values** option in the **CONTOUR** menu.

9.10.7.7. Closure Calculation Time

Closure calculation time

Despite bucket-sorting the closure calculation is still numerically intensive, and can take some time to perform. Therefore the following strategy is adopted:

- The closure calculation is only carried out when actually required. For example when a plot is requested, or scalar data extracted
- Once calculated the results are stored, so that subsequent plots, changes of contour values, etc, are fast as no recalculation is required.

The closure calculation has to be repeated when:

- Either the workpiece or die contents are changed, by adding or removing elements.
- Any closure parameter (thickness, bucket size, etc) is changed.
- A new complete state is read in.

- A velocity arrow or contact force vector plot is carried out. (Since the temporary storage space used is overwritten by these two plotting modes.)

While closure is being calculated a "progress" bar is written to the dialogue box stating how far, in %age terms, the calculation has gone. You can use **STOP** to terminate it prematurely at any time.

9.10.8. VISUALISATION

VISUALISATION

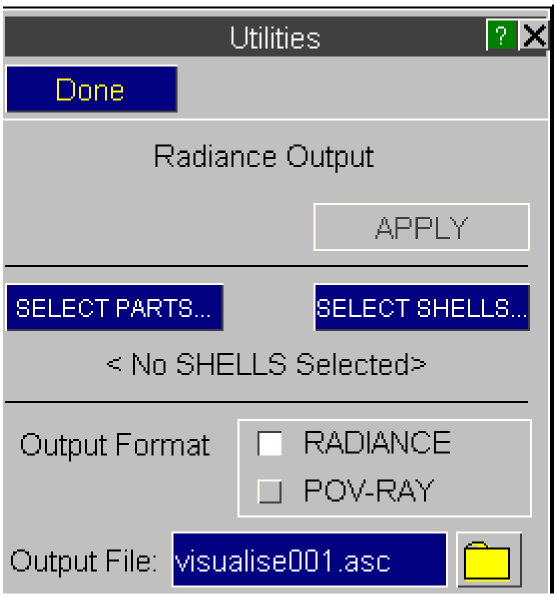
The figure (right) shows the **VISUALISATION** control panel. This option allows input files for ray tracing codes to be generated.

Output files may be written in a format suitable for either **RADIANCE** or **POV-RAY** .

Notes

1) Only shell elements/materials may be selected for output.

2) ASCII files generated using this option can become very large due to the input formats required by **POV-RAY** and **RADIANCE** .



9.10.9. UTILITIES, SETTINGS_FILE

UTILITIES, SETTINGS_FILE

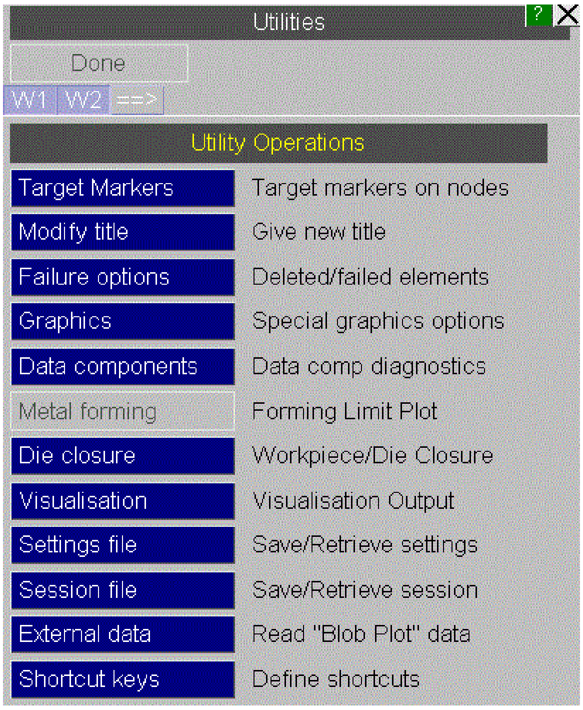
The "settings" file (extension `.set`) stores programme and window status information.

Its purpose is to allow you to define a screen layout that is model-independent, and to revert to that layout with minimum effort.

The settings file stores the number of graphics windows, and for each window:

- The current view.
- The background, text and hidden infill colours.
- The current **ENTITY** and **LABEL** settings.
- The current data component information and plotting mode.
- The current contouring settings and colours.
- Any cut-section data.
- Any fixed node, shift deformed and reference data.
- Lighting and shading information.
- Most **DISPLAY_OPTIONS** settings.
- Other miscellaneous data.

Reading in a settings file will optionally open further windows if the number currently present is less than that in the file, then apply the stored settings to all the windows in the display.



9.10.9.1. The Settings File (.set) Stores Programme, Not Model, Information

The settings file (.set) stores Programme, not Model, information

Almost all the values in the settings file are model-independent, and can be applied to any analysis. The exceptions are:

The current view.	Both the direction cosines (the view direction) and the current centre, scale and perspective distance are stored. If applied to a model with significantly different dimensions it may be necessary to **AC** (autoscale) after reading the settings file in order to make the model visible.
Nodes used in **Fixed node** , **Shift Deformed** and **Relative Values** .	If any of these nodes are defined they are converted to their external labels when written to the settings files. When the file is read back in an attempt is made to match the external label in the current model. If this is successful (for all nodes in a context where > 1 are required) that mode is restored, otherwise that mode is switched off.
Nodes used in **Cut_Sections** , and " **Cut follows nodes** "	Likewise nodes used to define a cut-section are written using external labels. When reread an attempt is made to match these in the current model. For cut section ***definitions*** it doesn't matter if the node(s) are not matched, since they are only used to derive section geometry and can be dispensed with once this has been computed. However if " **Cut follows nodes** " has been selected, but any of the nodes cannot be matched, this mode will be switched off.

The Properties (.prp) file stores Model-specific information

Information about models: item colour, transparency, brightness, overlay, labelling, etc is stored in the "properties" file. More details about this file are given under **PROPS** in Properties: Controlling colour, drawing style, transparency, lighting attributes and overlay of entities .

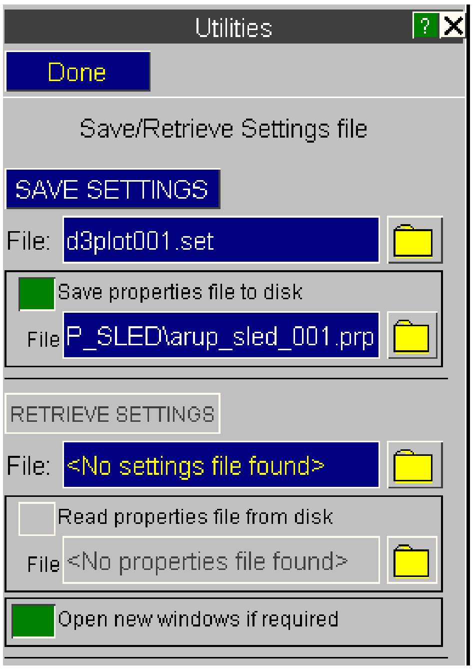
Properties files are written using external labels, so they can be applied to different models. Where an external label is not matched in the current model, or no counterpart to a current item is found in the properties file, no action is taken. Therefore properties files can be used to restore model settings to broadly similar analyses.

However the Settings and Properties files are closely related

Because these files are so closely related in terms of what they do it makes sense to process them together, and the **SETTINGS_FILE** panel allows you to read and write both types.

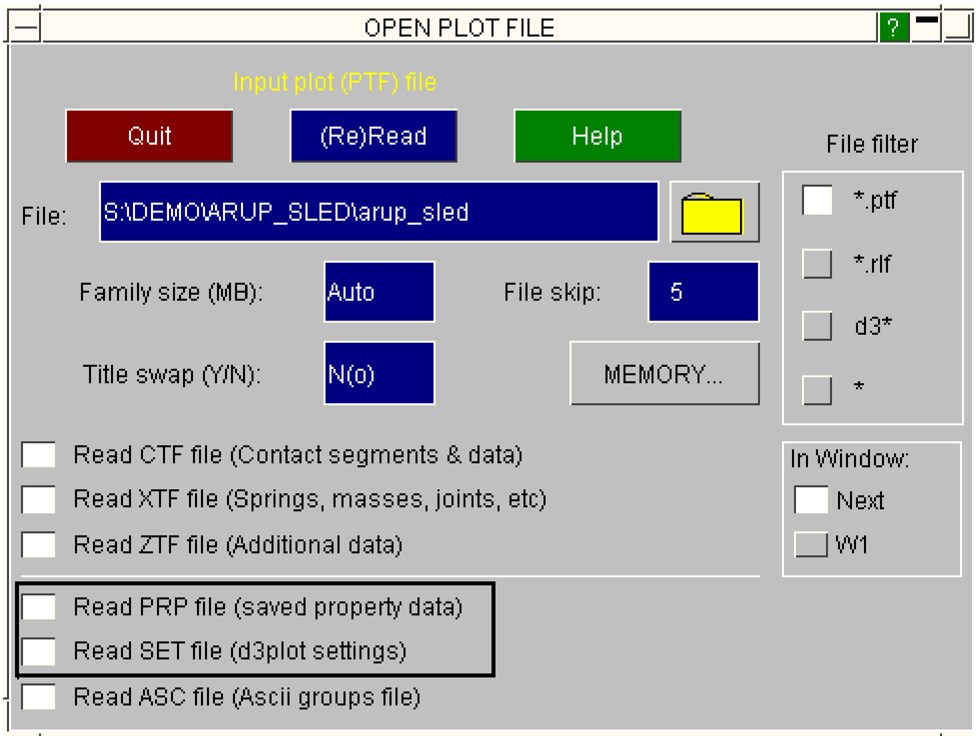
The SETTINGS FILE Panel

SAVE_SETTINGS	Stores the settings of all current windows in the specified " `.set` " files. The default filename " `d3plot` ***`nnn`*** `.set` " is recommended, although any filename may be used. If **Save properties file to disk** is selected then the specified properties file is also written. Any filename may be used, but the extension " `.prp` " is recommended.
RETRIEVE_SETTINGS	Reads in the specified settings file. If several are found the most recent is shown. If **Read properties file from disk** is selected the specified properties file will be read. In this example no such file has been found, so this option is not available. If **Open new windows if required** is selected then, if the settings file contains information for more windows than are current, additional windows up to this limit will be opened. Otherwise settings will only be applied to existing windows. If more windows are current than are defined in the settings file, then the stored values for window #1 will be applied to the excess ones.



9.10.9.3. Reading Settings and Properties Files With a New Model

Reading Settings and Properties files with a new model



On the **New Model** panel you can choose to read any Settings or Properties files present after you have read in the model.

The default filenames used will be:

The most recent Properties file in the model directory of the name:

<filename> _*nnn* .prp

The most recent Settings file in the current directory of the name:

d3plot *nnn* .set

These will be applied as if they had been read in explicitly above using **RETRIEVE_SETTINGS** .

9.10.10. UTILITIES, SESSION_FILE

UTILITIES, SESSION_FILE

The "session" file (extension `.dsf`) stores results file information, programme and window status information, and model properties.

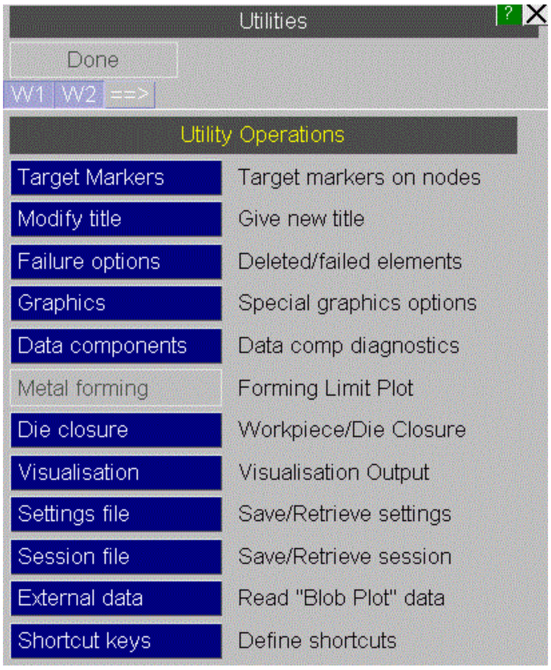
Its purpose is to allow you to save a particular D3PLOT state and to then quickly restore D3PLOT back to said state at a later time by loading a single file.

Session files may also be loaded while choosing results from a different iteration. Clearly, the results need to be similar to the data that is embedded in the session file in terms of geometry and entity list and numbering.

The session file stores the number of graphics windows, and for each window:

- State selection for XY data, envelope plots.
- The current view.
- The background, text and hidden infill colours.
- The current **ENTITY** and **LABEL** settings.
- The current data component information and plotting mode.
- The current contouring settings and colours.
- Any cut-section data.
- Any fixed node, shift deformed and reference data.
- Lighting and shading information.
- Most **DISPLAY_OPTIONS** settings.
- Other miscellaneous data.

Session files may only be loaded into an empty D3PLOT session where no models are pre-loaded.



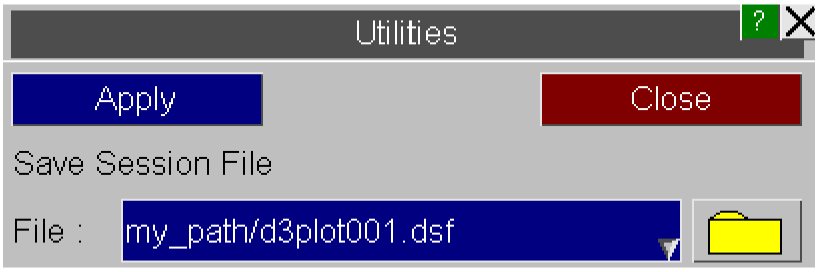
9.10.10.1. Saving Sessions

Saving sessions

Saving via the 'Session file' panel.

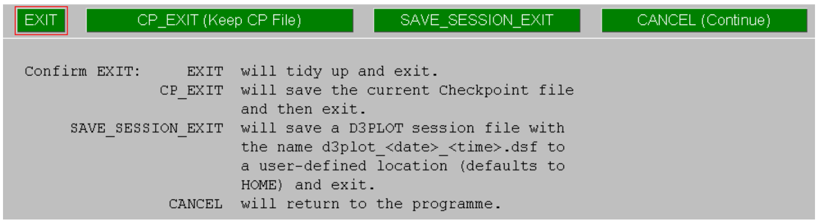
Stores all relevant information in one session file (" **`.dsf`** ").

The default filename " **`d3plot`** ***`nnn`*** **`.dsf`** " is recommended, although any user-defined filename may be used.



Saving via the SAVE_SESSION_EXIT option during exit

Clicking **SAVE_SESSION_EXIT** will save a session fle with a default name with the format **`d3plot_<date>_<time>.dsf`** . The file will be saved in the home directory by default but the location can be controlled using preferences.



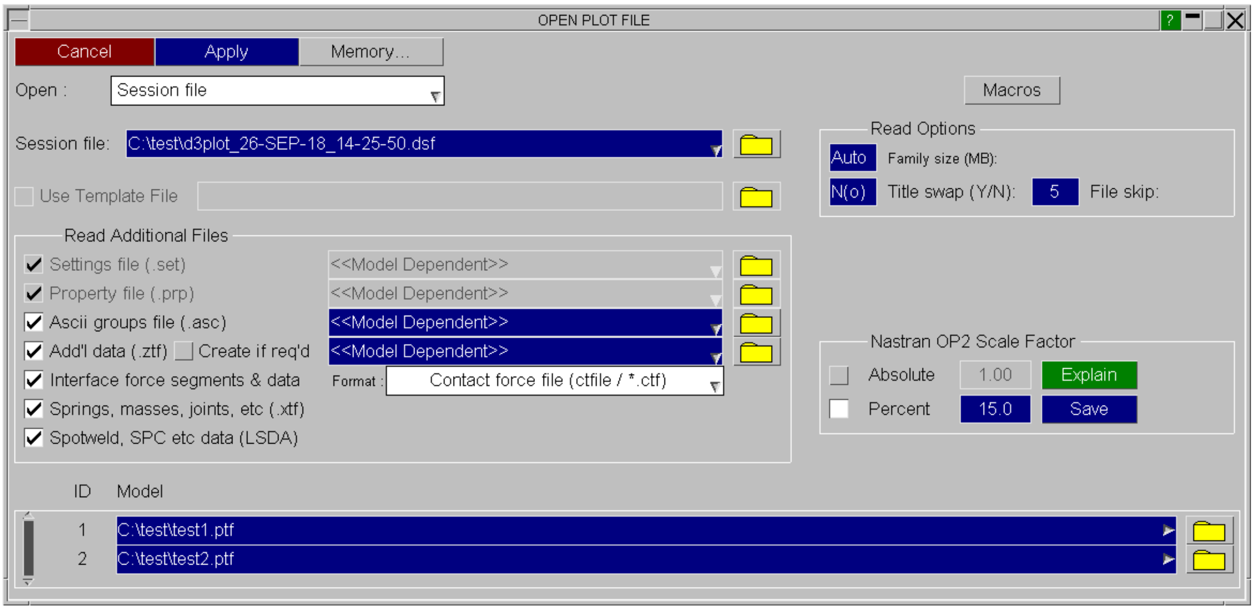
Saving a session unconditionally upon exit

The preference **`d3plot*session_auto_save`** , when set to ON, will automatically write out a session file every time D3PLOT exits. These automatic session files will carry the name **`d3plot_<date>_<time>.dsf`** .

The save location may be controlled by using the preferences **`d3plot*session_save_option`** and **`d3plot*session_save_dir`** .

9.10.10.2. Reading Session Files in a New Session

Reading Session Files in a new session



On the **New Model** panel you can choose to read any Session File. This must be done before reading in a model..

The most recently saved session file names can be accessed using the appropriate recent files popup. Both explicitly saved sessions and auto-saved sessions will be listed.

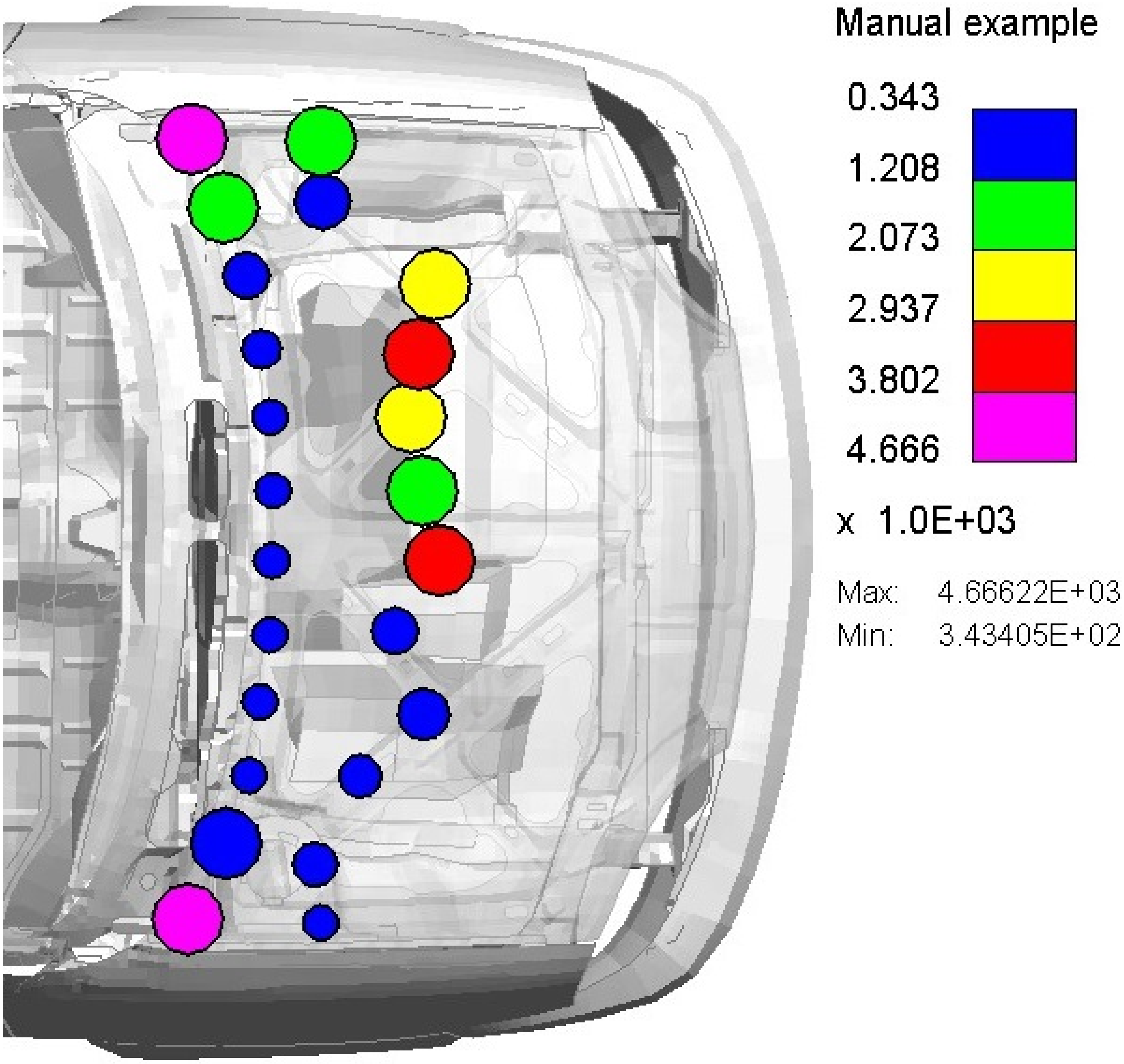
Result file names may be changed using appropraite textboxes or file selectors in order to load results from alternate iterations.

9.10.11. External Data

External Data

"External" data is arbitrary data at three-dimensional points, generated externally, that D3PLOT can superimpose on the current image in a variety of ways.

The following - artificial - example shows notional head-impact values on a car bonnet, but any data from any source can be used.
(External data plots are sometimes referred to as "Blob plots", for reasons that are obvious from the plot below!)



9.10.11.1. How External Data Plotting Works

How External Data plotting works

An "external data" file is generated (externally) by the user, by any means, that contains some or all of the following attributes:

- Arbitrary data values at three-dimensional coordinates in space
- Contour band values and colours
- Information about symbol size and labelling
- Lines used to demarcate regions

D3PLOT reads this file, supplying defaults for information that is not specified, and displays the results on top of the current image. This display can be turned on or off at will, and modified at any time.

External data is stored on a "per model" basis, so if you have multiple models data has to be read separately for each model and its visual attributes may also be controlled separately.

9.10.11.2. The Format of an "External Data" File

The format of an "External Data" file

External data files are free format ASCII files in which data can appear in any order (unless stated otherwise below). They may contain any or all of the following information:

Header	Data type	Data format example
`$` xxx and blank lines	Lines starting with " **$** " and blank lines are treated as comment lines, they can appear anywhere in the file and are ignored	`$ This is a comment line`
name < ***data name*** >	A title of up to 30 characters	`name This is a demonstration example` Will read the title "This is a demonstration example"
`cont_format <format>`	Denotes how the number format of the contour bar is determined. <format> can be 'auto', 'manual', 'scientific' or 'general'.	`cont_format manual` Will set the contour bar number format to manual
`cont_exp <exp>`	The exponent number to use for a manually defined contour bar number format.	`cont_exp 3`

		Will set the exponent number to E+ 3
`cont_dp <dp>`	The number of decimal places to display on a manually defined contour bar number format.	**`cont_dp 2`** Will set the number of decimal places to 2
`automatic < #bands >.`	Denotes automatic contours to be used with <#bands> levels where 1 <= #bands <= 13 Contour bounds are determined by scanning the data once read.	**`automatic 10`** Will use automatic contour levels with 10 bands

`levels < #levels >`
 	
`< L1> <rr> <gg> <bb>`
`<L2> <rr> <gg> <bb>`
to #levels lines
	Denotes manually specified contours with <#levels> values where:
	
	2 <= #levels <= 14
	
	Each level has a value <Ln> followed by Red Green Blue colour components, each component being in the range 0 - 100.
	
	Colour <i> spans the band <i> to <i+1>
		`levels 6`
		`0 0 0 100`
		`5 0 100 100`
		`10 0 100 0`
		`15 100 100 0`
		`20 100 0 0`
		`25 100 0 100`
		
		Sets up 6 levels, ie 5 bands, from blue to magenta

Note on contour levels / bands:

- Commands " **`automatic`** " and " **`levels`** " are mutually exclusive, you cannot have both.
- If neither is defined then the default is " **`automatic`** " with the same number of levels as currently used for normal contouring.
- In the " **`levels`** " case you must follow the levels command with the correct number of lines of data.

circle_f < ***diam*** > **circle_v** < ***factor*** > < ***min_dia*** > < ***max_dia*** > **rect_f** < ***width*** > < ***height*** > **rect_v** < ***fac_w*** > < ***fac_h*** > < ***min_dim*** > < ***max_dim*** >	Only one of these options may be used. • **circle_f** is a fixed diameter circle • **circle_v** is a variable diameter circle, with diameter set to < ***factor* * *data*** > • **rect_f** is a fixed size rectangle • **rect_v** is a variable size rectangle of width < ***fac_w* * *data*** > and height < ***fac_h* * *data*** > In both "variable" cases there is limiting minimum and maximum size for the symbol. If nothing is defined the default is a fixed size rectangle of 20 x 20 units.	**circle_v 0.1 30 250** Would set a variable diameter circle based on 0.1 * the incoming data value, and subject to a minimum size of 30 units and a maximum of 250 units. **rect_v 0.2 0.1 100 300** Would set a variable sized rectangle, width = 0.2 * data, height = 0.1 * data, with a minimum size of 100 units and a maximum of 300 units. Screen units are based on the scaled screen unit of 4096 units across the window width.
show_value <true/false>	Whether or not the data value is drawn as a label on top of each point's symbol. The default is **false** .	**show_value true** Would cause values to be drawn.
show_nid <true/false>	Whether or not the node id is drawn as a label on top of each point's symbol (for points defined with <ndata>). The default is **false** .	**show_nid true** Would cause node ids to be drawn.
show_coord <true/false>	Whether or not the x,y,z coordiantes are drawn as a label on top of each point's symbol (for points defined with <data>). The default is **false** .	**show_coord true** Would cause coordinates to be drawn.
show_text <true/false>	Whether or not arbitrary text is drawn as a label on top of each point's symbol. The default is **false** .	**show_text true**

		Would cause arbitrary text to be drawn.
`data <x,y,z> <value> <text>`	Data point values. Each point must have an (x,y,z) coordinate followed by a value. The value may be floating point or integer. Some optional arbitrary <text> (limited to 80 characters) can be written at the end of the line to annotate the data. If there are spaces in the text it needs to be enclosed by " ". Any number of data values may be input, by default none is defined.	**`data 10.1 20.2 30.3 500.0 "some text"`** Would make a point at position (10.1, 20.2, 30.3) with a value of 500.0 and 'some text' will be displayed if the show_text flag has been set to true (see above).
`ndata <node_id> <value> <text>`	Defines a data <value> at node <node_id>. The data value will remain constant, but its plotted position will always be the node's current coordinate at a given time. Some optional arbitrary <text> (limited to 80 characters) can be written at the end of the line to annotate the data. If there are spaces in the text it needs to be enclosed by " ". Any number of data values may be input, by default none is defined.	**`ndata 101 350.0 "some text"`** Would specify a value of 350.0 at node 101 and 'some text' will be displayed if the show_text flag has been set to true (see above)
`beam <x1,y1,z1> <x2,y2,z2> <colour>`	Defines a "beam" (really just a line) from (x1,y1,z1) to (x2,y2,z2) in colour number <colour>. At present colours are hard-wired as follows:	**`beam 1.0 1.0 1.0 2.0 2.0 2.0 2`** Would drawn a medium width green line from (1,1,1) to (2,2,2)

	1. Red, thin line 2. Green, medium width line 3. Blue, thick line (This feature is a crude solution and will probably develop in the future.)	

9.10.11.3. Reading and Controlling External Data Files

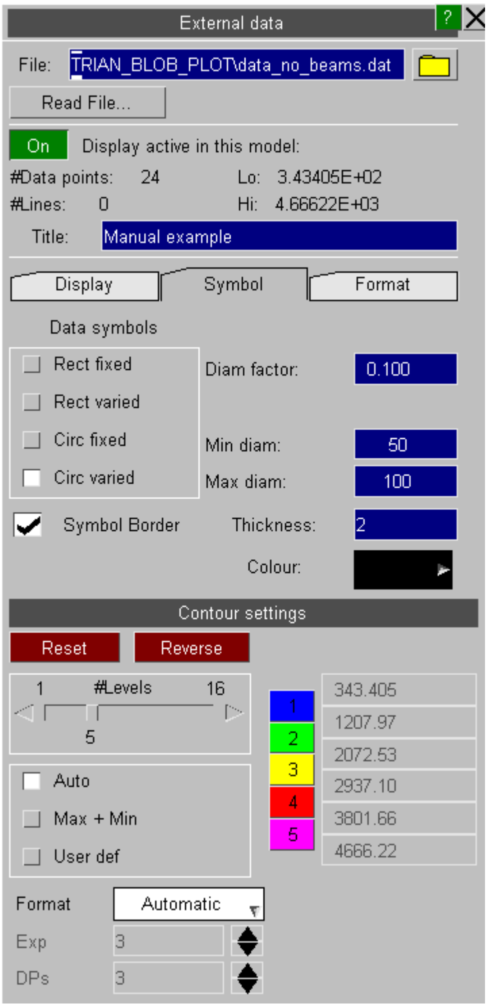
Reading and controlling External Data files

Utilities, External Data maps the panel shown here.

Use the **File:** prompt to define the external data filename, then **Read File...** to read it in and store it in memory.

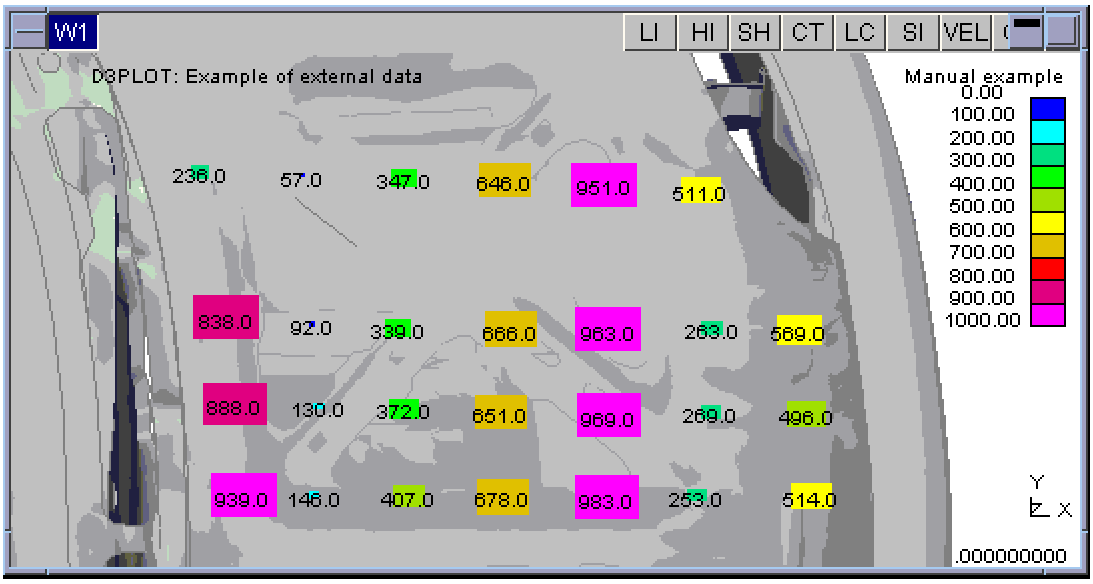
Thereafter the key parameters described above (contour bands, symbol type and size, data value display) will be used to initialise this panel, but you can change them interactively at will.

The turning off/on of display of external data on plots is controlled by **Display active in this model.**

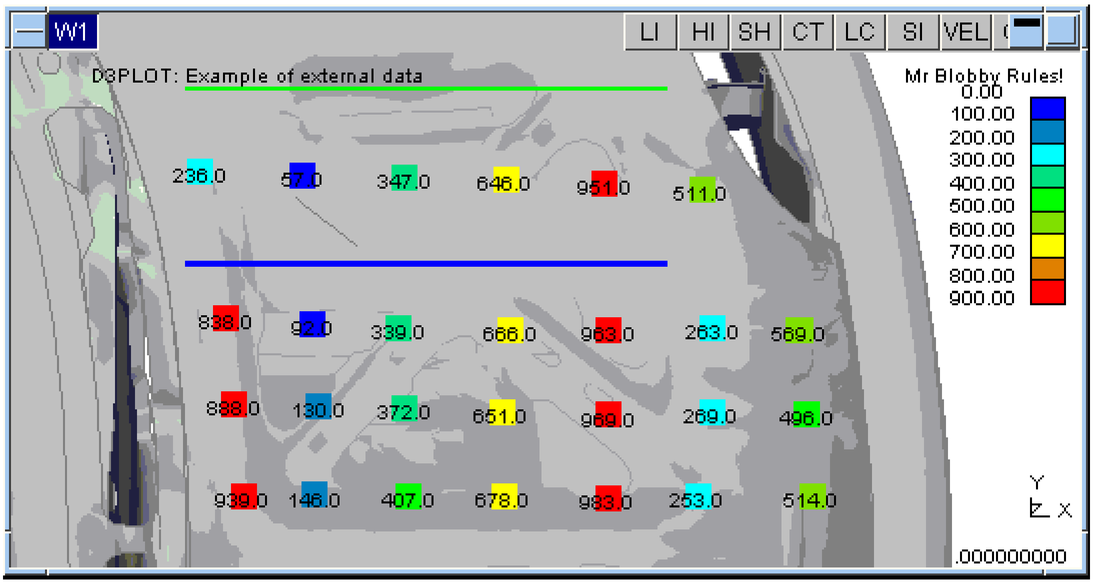


Some examples. Note that displaying the underlying model in grey usually works well as it gives good contrast with the coloured symbols, as does using transparency to show underlying structure.

Here the symbols in the plot above have been changed to rectangular, varying; and display of data values has been turned on.



Here symbols have reverted to fixed size squares, and two "beams" have been added to demonstrate their appearance.



9.10.12. UTILITIES, FUNCTION KEYS

UTILITIES, FUNCTION KEYS

It is possible to programme the function keys F1 .. F12 with command files.

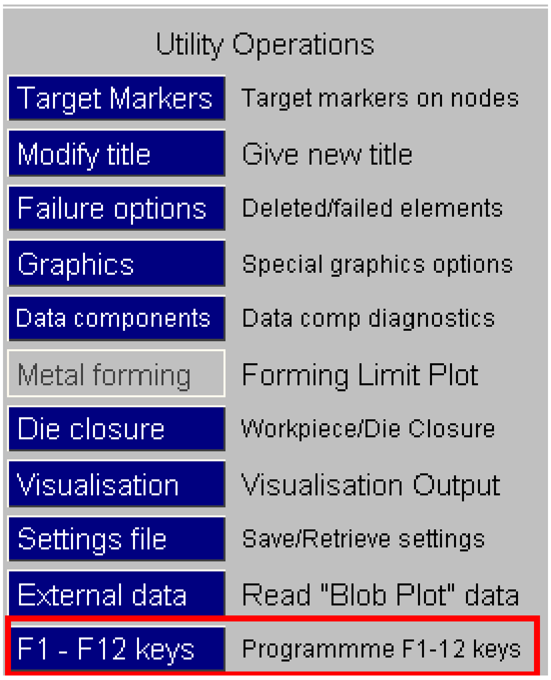
- You create a command file in the normal way.
- The file is associated with a function key.
- It is executed whenever you press that key.

A command file can contain any valid sequence of D3PLOT commands, for example blanking some items and setting a view, or defining a cut-section. Generally it is best to avoid very model-specific commands, or screen picking, if a file is to have a general usefulness on playback. (Command files are described in Appendix G.)

Files can be associated automatically with function key " ***n*** " (1 <= ***n*** <= 12) as follows:

- By the "oa_pref" file line **"d3plot*f *n* _key: <** *filename* **>"**
- By defining file **d3plot_f *n* .tcf** in the current directory (**$cwd)**
- By defining file **d3plot_f *n* .tcf** in your home directory (**$HOME**)
- By defining file **d3plot_f *n* .tcf** in the **$OASYS** directory

Files are searched for in the order given above, and the first definition found "wins". If no definition is found that function key will be inactive.



9.10.12.1. Using the Function Key Panel

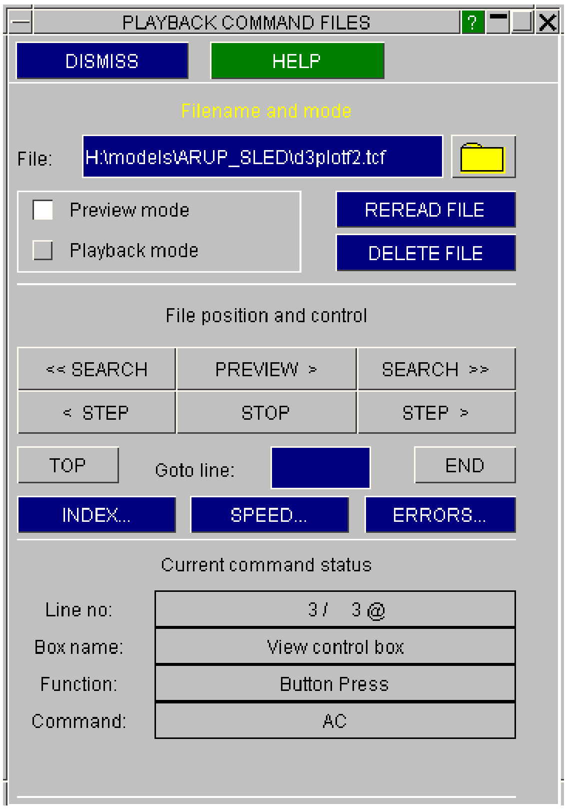
Using the function key panel

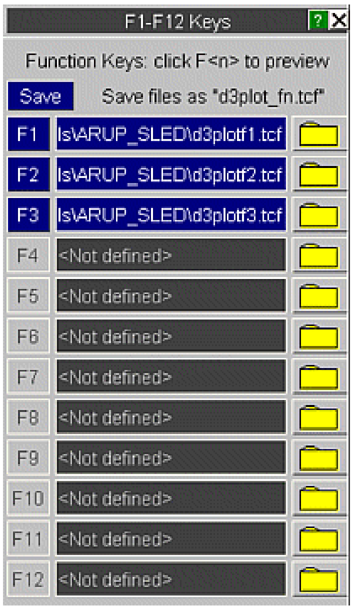
The function key panel shows the current status of all function keys.

You can associate a new command file with a function key at any time by typing in new filename, or use the "?" file filter to find new keys.

SAVE... copies all current definitions to d3plot_f1.tcf ... d3plot_f12.tcf in the current directory. These local copies make it easier to modify standard files for your own use.

Clicking on the **F1** to **F12** buttons will Preview the command file in the standard playback panel. This simply steps through the file's contents without actually executing it:





9.10.12.2. Function Key F10 is Reserved on Windows Platforms

Function key F10 is reserved on Windows platforms

On Windows platforms the F10 key is reserved by the Windows operating system and cannot be used for D3PLOT macros.

You can still define a function for key F10 in the oa_pref file, so files set up on Unix or Linux machines will still read in, but the key will not work and it will be necessary to associate the macro with a different key.

A warning is issued if you attempt to define a macro for the F10 key on these platforms.

9.10.13. COMPRESS

COMPRESS

This option can be used to generate a new set of PTF/D3PLOT files for a model that contain only a subset of the Parts and States in the original model. See [WRITE] Cutdown D3PLOT/PTF file for more details.

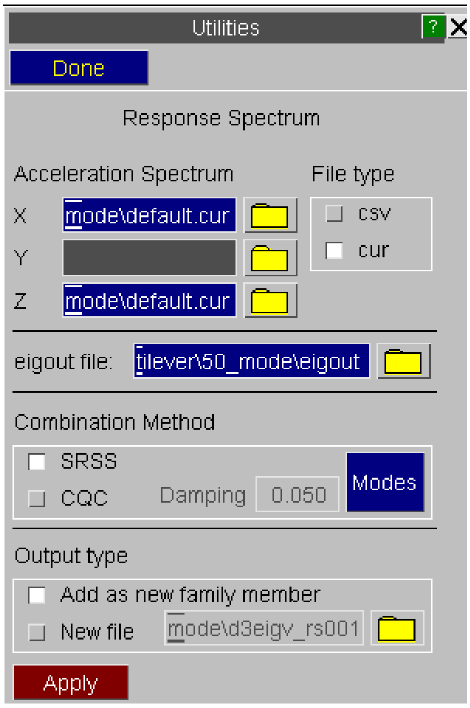
9.10.14. UTILITIES, RESPONSE SPECTRUM ANALYSIS

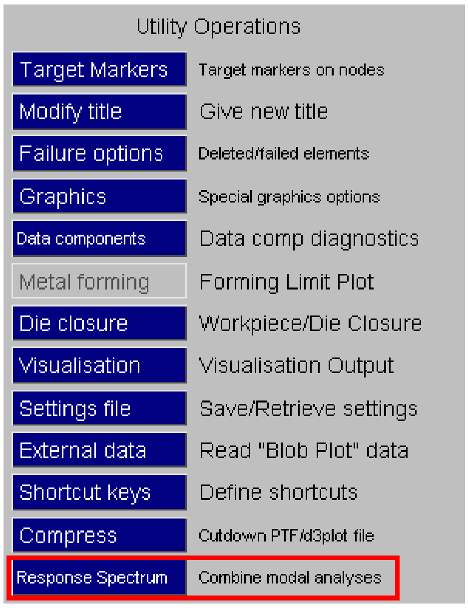
UTILITIES, RESPONSE SPECTRUM ANALYSIS

The **RESPONSE SPECTRUM ANALYSIS** panel is intended for use in a seismic analysis to combine the multiple modes of a structure into one mode, so that the total response of the structure can be seen. Two methods of combination are available in D3Plot; the Square Root Sum of Squares (SRSS) and the Complete Quadratic Combination (CQC) method.

In order to calculate the combined mode, the user needs to supply the following information in the pane, shown below:

1. Time Period vs. Acceleration curves, for at least one translational degree of freedom, defining the acceleration spectrum. See RESPONSE SPECTRUM ANALYSIS Time Period vs. Acceleration curves .
2. The eigout output file produced by Dyna, containing the participation factors for each mode. See RESPONSE SPECTRUM ANALYSIS Eigout File .
3. The combination method to use. See RESPONSE SPECTRUM ANALYSIS Combination Methods .
4. How the new mode is output. See RESPONSE SPECTRUM ANALYSIS Output





9.10.14.1. RESPONSE SPECTRUM ANALYSIS Time Period vs Acceleration Curves

RESPONSE SPECTRUM ANALYSIS Time Period vs. Acceleration curves

The top part of the **RESPONSE SPECTRUM ANALYSIS** panel is where the Time Period vs. Acceleration spectra curves are input. A curve must be specified for at least one of the translational degrees of freedom.

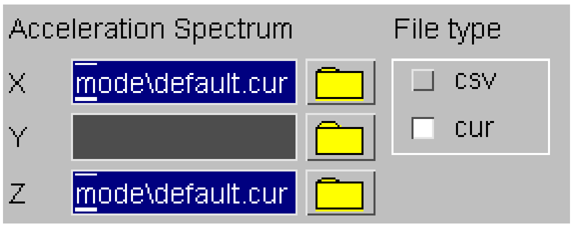
The curves may be either *.csv or T/His curve files. The expected format of the *.csv file is:

```
<Title>
<x-axis label>,<y-axis label>
<TimePeriod_1>,<accn_1>
<TimePeriod_2>,<accn_2>
. .
. .
<TimePeriod_n>,<accn_n>

The expected format of the T/His curve file is:

$
$ Comments
$ (as many lines as you want starting with a $)
$
<Title>
<X-label>
```

```
<Y-label>
<Curve name>
<TimePeriod_1> <accn_1>
<TimePeriod_2> <accn_2>
. .
. .
<TimePeriod_n> <accn_n>
```

If the formats differ from this, the curves will not be read correctly.

Note: If the time period of a mode is outside the bounds of the time periods defined in the curves then the nearest point is used. A warning is printed to the dialogue box if this happens.



9.10.14.2. RESPONSE SPECTRUM ANALYSIS Eigout File

RESPONSE SPECTRUM ANALYSIS Eigout file

To combine the modes, D3Plot needs to know their participation factors. This information is contained in the eigout file produced by Dyna.

9.10.14.3. RESPONSE SPECTRUM ANALYSIS Combination Methods

RESPONSE SPECTRUM ANALYSIS Combination Methods

D3Plot provides two methods for combining the modes; Square Root Sum of Squares (SRSS) and Complete Quadratic Combination (CQC).

Square Root Sum of Squares

For each result value stored in the ptf file, the SRSS method calculates the combined total with the formula:

$$\sqrt{\sum_{i=1}^{n}\left[\left(f_{part_{ix}}\,a_{spectral_{ix}} + f_{part_{iy}}\,a_{spectral_{iy}} + f_{part_{iz}}\,a_{spectral_{iz}}\right)\frac{mode\_mass\_factor * value_i}{2\pi f_i}\right]^2}$$

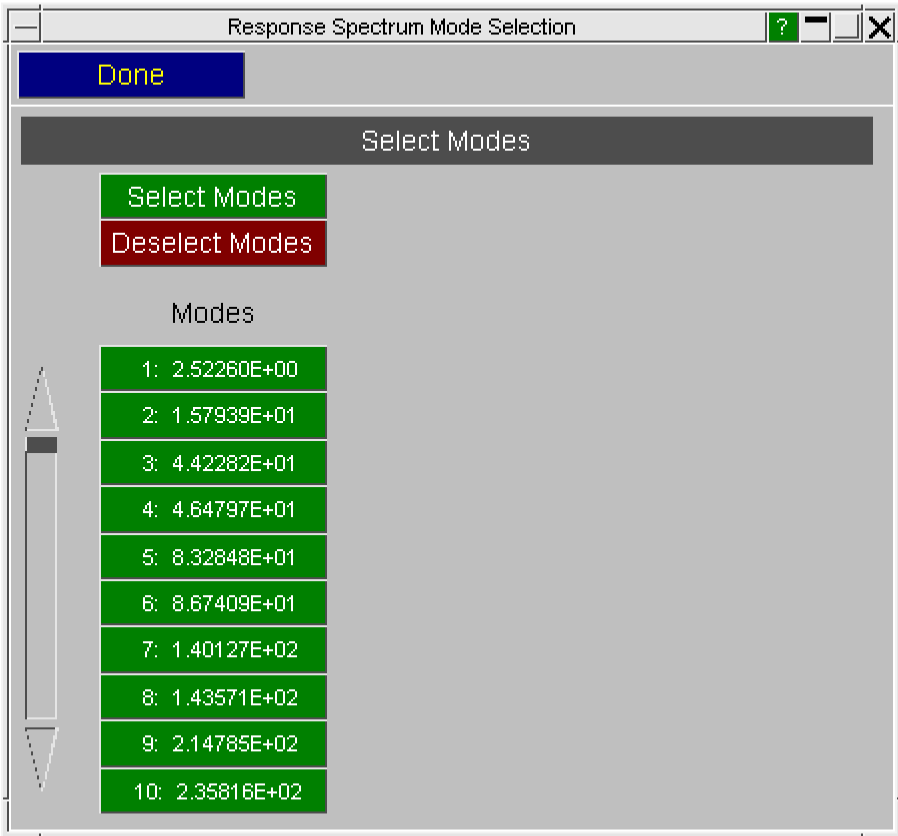
Where:

	Are the participation factors for mode <i> in the global X, Y and Z directions. These are extracted from the 'eigout' file output from Dyna.
$a_{spectral_{ix}}$, $a_{spectral_{iy}}$ and $a_{spectral_{iz}}$	Are the spectral accelerations for mode <i> in the global X, Y and Z directions. These are extracted from the Time Period vs. Acceleration curves provided by the user.
f_i	Is the frequency of mode <i>.
$mode\_mass\_factor$	Is a factor that is applied to the results by Dyna to make the mode shapes visible. This is extracted automatically by D3Plot from the results file.
$value_i$	Is the results value at mode <i>

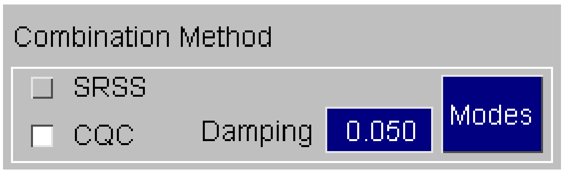
There are two important points to note:

1. All results values are processed this way. This will not make sense in some cases, for example, thin shell thickness will be incorrect in the combined mode.
2. Thin shell stress tensors are rotated to the elements local coordinate system of the undeformed geometry before they are combined. They are rotated back to the global coordinate system in the combined mode.

By default the results from all the modes will be combined, but if you wish you can choose to only consider a subset of modes. To do this, press the **Modes** button to bring up the panel shown on the right.



Complete Quadratic Combination



The CQC method is an improvement on the SRSS method as it allows for interactions between closely spaced nodes, but it takes longer to process.

For each result value stored in the ptf file, the CQC method calculates the combined total with the formula:

$$\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} q_i \rho_{ij} q_j}$$

Where:

$q_i = (f_{part_{ix}} a_{spectral_{ix}} + f_{part_{iy}} a_{spectral_{iy}} + f_{part_{iz}} a_{spectral_{iz}}) \frac{mode\_mass\_factor * value_i}{2\pi f_i}$ Is the response for mode <i>

$$q_j = \left(f_{part_{jx}} a_{spectral_{jx}} + f_{part_{jy}} a_{spectral_{jy}} + f_{part_{jz}} a_{spectral_{jz}}\right) \frac{mode\_mass\_factor * value_j}{2\pi f_j}$$

Is the response for mode <j>

$$\rho_{ij} = \frac{8\sqrt{\xi_i \xi_j}\,(\xi_i + r_{ij}\xi_j)\, r_{ij}^{3/2}}{(1 - r_{ij}^2)^2 + 4\xi_i \xi_j r_{ij}(1 + r_{ij}^2) + 4(\xi_i^2 + \xi_j^2) r_{ij}^2}$$

Is the Cross Modal Coefficient for modes <i> and <j>

Where:

ξ_i and ξ_j Are the damping ratios for modes <i> and <j>

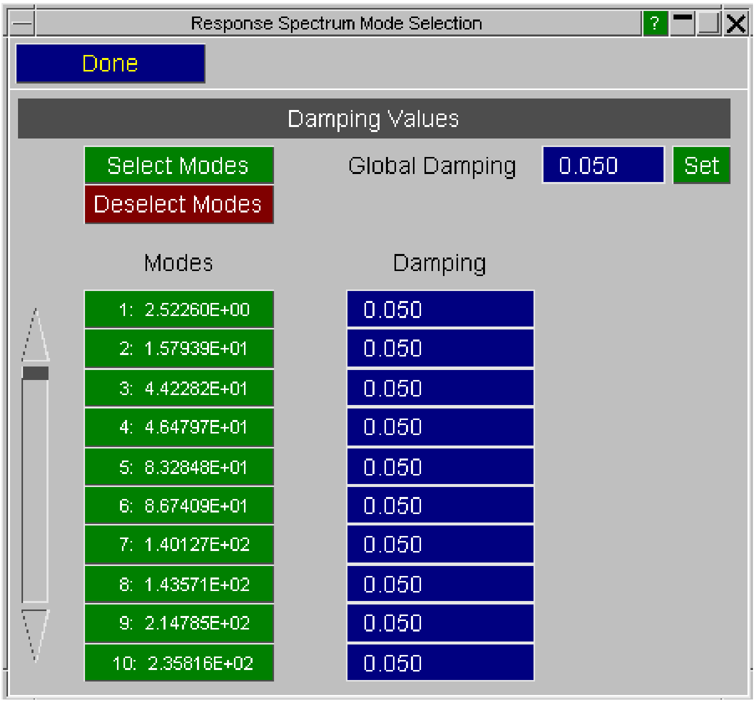
$r_{ij} = \frac{f_i}{f_j}$ Is the ratio of the frequencies of modes <i> and <j>

As with the SRSS method the following two points should be noted:

1. All results values are processed this way. This will not make sense in some cases, for example, thin shell thickness will be incorrect in the combined mode.
2. Thin shell stress tensors are rotated to the elements local coordinate system of the undeformed geometry before they are combined. They are rotated back to the global coordinate system in the combined mod

By default all modes will be processed and the damping for each mode is set to 5%. As with the SRSS method you can chose to select only a subset of modes to consider by pressing the **Modes** button. This will bring up the panel shown on the right.

Also in this panel different damping values can be set for each mode.



9.10.14.4. RESPONSE SPECTRUM ANALYSIS Output

RESPONSE SPECTRUM ANALYSIS Output

The combined mode can be output two different ways, either in a new family member of the opened model or as a completely new file. In both cases the combined mode will have a frequency value of 10*the highest mode in the model.

A log file will also be produced in the folder where the model resides called "resp_spec.txt". This contains information about which input files were used, the combination method, and the factors calculated by D3Plot to produce the combined mode.

9.10.15. UTILITIES, COARSEN

UTILITIES, COARSEN

Mesh coarsening is designed to improve the graphics speed of large and finely meshed models. It works by grouping adjacent elements into "patches" and hence sending fewer rendering requests down the graphics pipeline.

It results in a slight loss of image quality, but this is usually acceptable and by default D3PLOT reverts to the original (fine) mesh when you zoom in to look at details.

By default (i.e. if you don't use "Custom" settings) coarsening does the following:

- Only polygon fill (i.e. shaded, "fuzzy" shaded image and hidden infill) is coarsened.
- Wireframe and hidden overlay are not coarsened.
- Coarsening is based on the last state in the analysis.
- Zooming in by a factor of more than 4x the auto-scaled scale reverts to using the original mesh.

Coarsening is fully automatic, with no user intervention required unless you want to change the default settings.

By default Coarsening is not active, but you can modify this with the **`d3plot*auto_coarsen`** "oa_pref" file option so that large models are automatically coarsened when opened. See Appendix B for more information.

9.10.15.1. Coarsening Levels and what they Mean

Coarsening Levels, and what they mean

Off This is the default state. No coarsening takes place and the original mesh is used. Mild Coarsening takes place to a reasonable degree, usually with only mild visual artefacts appearing. Typical speed-ups of the order of 40% - 50% Severe More aggressive coarsening is used, usually with significant visual artefacts, however larger models can double in speed or better. Custom Allows user control over all the coarsening parameters.

Mild coarsening is the recommended level for most models, as it tends to give the best trade-off between speed and image quality.

9.10.15.2. Examples of Coarsening

Examples of Coarsening

The images below show a model with approximately 1,000,000 shell elements in its raw state, and illustrate how progressively harsher coarsening reduces the image quality but speeds up the graphics.

This is the original image. Time to animate a frame on representative hardware: approx **95mS / frame**

This is the same model using **Mild** coarsening. Some "speckling" is evident and the overlay is poking through in a few places. However time to animate this model is now reduced to about **50mS / frame.**

Finally here is the same model using **Severe** coarsening. More artefacts are visible, and the large patches are clearly evident on the glass panels. However time to animate now approx **30mS / frame** , which is over 3x as fast as the original model above.

9.10.15.3. Custom Coarsening

Custom Coarsening

Use Coarsened Mesh determines the scale at which the transition between true and coarsened mesh takes place.

Never	Means that the true mesh is used at all scales
If max elem size is < xxx pixels	Not recommended, as it requires a knowledge of screen resolution.
If scale factor < fff	This is the factor on the normal "auto-scaled" scale for this model. At factors higher than < **fff** >, i.e. when zoomed in, the original mesh will be used.
Unconditionally	The coarsened mesh is used at all scale factors.

Use for LI, HI and Overlay determines whether or not the coarsened mesh lines are used for wireframe plots, and overlay on shaded and contoured plots.

Normally you should leave this box unchecked, as the coarse overlay is not representative of the true mesh, however it can be useful when tuning some of the other parameters.

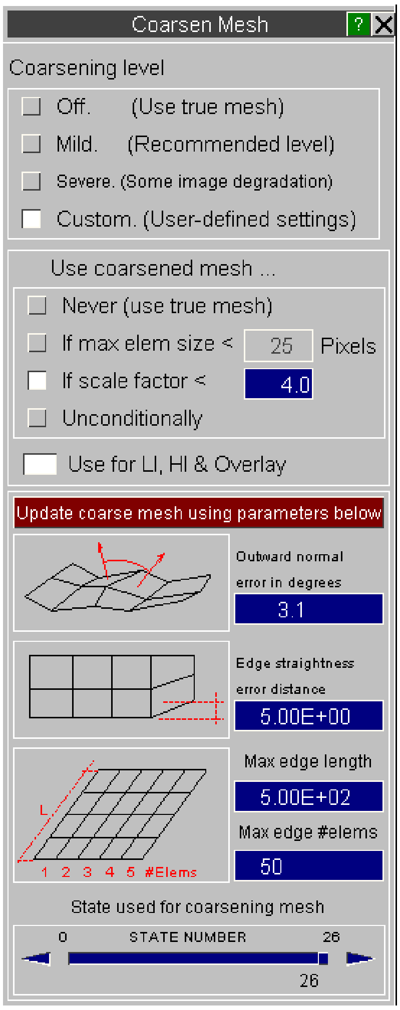
Outward normal error, Edge straightness error and **Max edge** attributes.

Play with these at your peril! The default values usually work reasonably well, but they have been arrived at by trial and error over a wide range of models rather than by any scientific means. It is beyond the scope of this manual to explain the theory behind them.

State used for coarsening mesh determines which state is used when extracting the coarse "Patches". An example of using this is given below.

Normally this should be the last state, as this will usually have the greatest deformation and hence limit the elements that can be merged into patches. However if the last state is corrupt, or you are only displaying earlier states, you may get an improvement in both image quality and speed if you choose an earlier state.

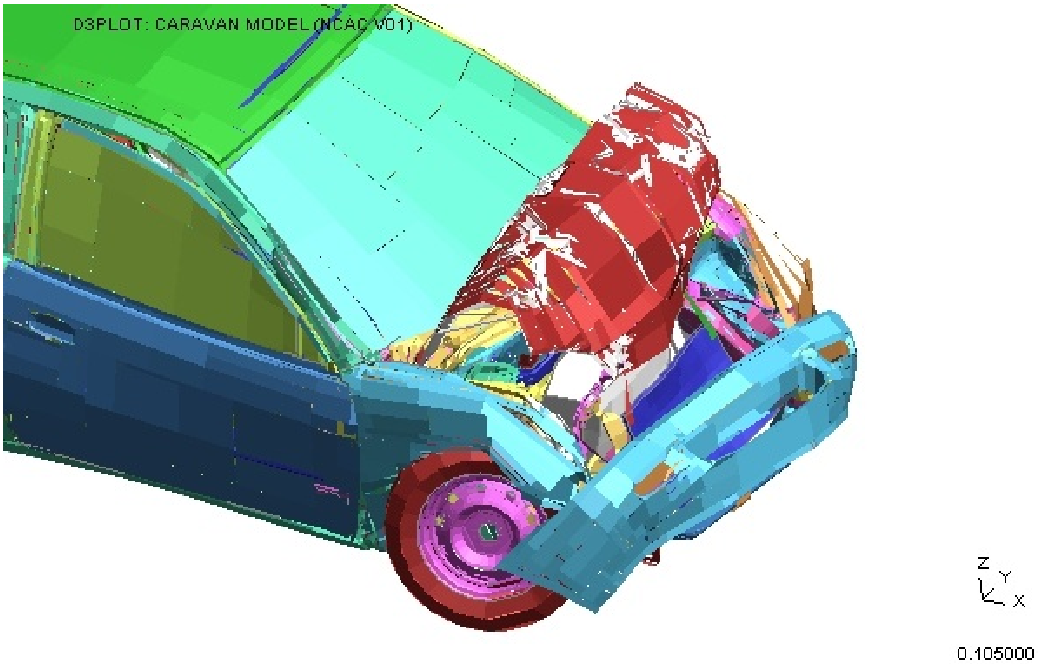
Update Coarse mesh using parameters below . If you modify any of the detailed parameters the coarse mesh will not be recalculated, and the changes will not therefore take effect, until you press this button.



9.10.15.4. What Happens when you Base Coarsening on an Earlier State

What happens when you base coarsening on an earlier state

In this image, using the same model as above, **Severe** coarsening has been used but has been based on the undeformed geometry at state #1. The errors that arise from basing patches on elements which subsequently become deformed are obvious!



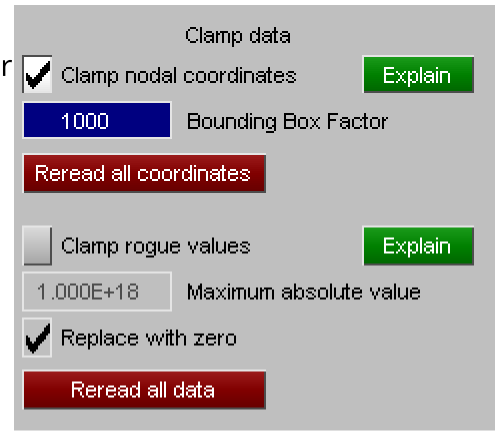
9.10.16. Clamp Data

Clamp Data

Restricting the displacements of nodal coordinates to prevent "shooting nodes" causing a problem

Also "clamping" the magnitude of data values generally to prevent rogue values obscuring valid data.

Clamp nodal coordinates



Generally displacements in a model are not large compared to the overall model dimensions, however it is sometimes the case that elements are deleted and their associated nodes become unrestrained. Isolated nodes have only minimal mass, so if any force continues to be applied to them they can rapidly acquire enormous displacements, a process which is generally referred to as the "shooting nodes" problem.

This can cause problems during post-processing since the auto-scaling process may consider these nodes if there is still something "structural" attached to them. Alternatively if the scale is reset to show the wanted parts of the model they can cause a freeze or even a crash in the graphics card as it tries to render items which are in a different galaxy at the current scale. *(This is not an exaggeration: 1 light-year is approximately 9.5e15 meters. A single precision floating point* *value has a maximum value of ~1e38, so even if a model uses mm units a node with this displacement value is around 1e20 light years away. There is some disagreement about the size of the universe, but our nearest galaxy, Andromeda, is "only" ~2.5e6 light-years away.)*

In order to try to prevent this problem D3PLOT 11 onwards now clamps nodal coordinate values to a Bounding Box Factor (a multiple) of the size of the bounding box round the model's undeformed geometry, with 1000x being the default value. Nodal coordinates outside this range are reset to the model's centre centre coordinate. A multiple has been chosen in preference to an absolute value since it relates the clamping value to the size, and hence units, of the model. Any multiple value >= 1 may be used.

This option is turned on by default, however both on/off switch and bounding box factor may be controlled by the following preferences:

`d3plot*clamp_nodes:`	**`TRUE`** or **`FALSE`**
`d3plot*clamp_node_factor:`	**`Integer >= 1`**

Reread all coordinates

The clamping process only takes place on nodal coordinates when they are read from disk into memory, meaning that any coordinates currently in memory will not be affected by a change to the bounding box factor. If you want to apply a revised factor to the current image then use **Reread all coordinates** to force D3PLOT to delete and reread all nodal coordinates afresh. This is carried out as follows:

- ***All*** existing nodal coordinates, ***for all models*** , are deleted from memory
- Coordinates are reread "on demand" as and when they are required - generally to render new images.

Clamp rogue values

A separate, albeit related, problem is that analyses that have gone wrong or "blown up" in some way can generate very large data values which can obscure wanted, typically much smaller, values by extending automatic contour bands to ridiculously large max/min values. D3PLOT V11 onwards can now detect and "clamp" these values to limit the effects of this problem. The process works as follows:

- Whenever floating point data is read from disk each value's magnitude is compared against the specified limiting magnitude.
- If it exceeds this value it is "clamped" either to that magnitude times its sign (i.e. -1e20 would be clamped to -1e18 in the default case), or to zero if **Replace with Zero** has been selected.

This setting is not turned on by default, otherwise it might obscure problems in a model - especially if **Replace with Zero** is in force. The default magnitude of 1e18 has been chosen because of the possibility that some data values might be squared during processing, for example to calculate a vector length, and 1e18 squared is 1e36 which gives a bit of protection against floating overflow given the single precision floating point limit of ~1e38.

This option can be controlled by the following preferences:

`d3plot*clamp_data:`	`TRUE` or `FALSE`
`d3plot*clamp_max_value:`	`Floating point value > 0.0`
`d3plot:clamp_to_zero:`	`TRUE` or `FALSE`

Reread all data

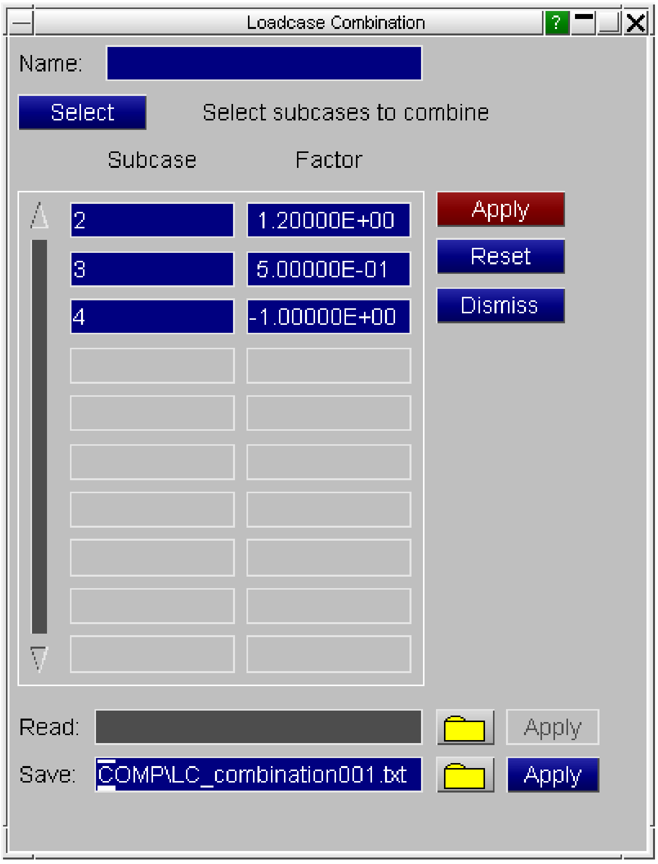
As with coordinates above data clamping is only applied when data are read from disk into memory, so to apply clamping retrospectively to values already in memory use **Reread all Data** which works as follows:

- ***All*** existing data (including nodal coordinates), ***for all models*** , are deleted from memory
- Data are reread "on demand" as and when they are required

9.10.17. Static Loadcase Combination

Static Loadcase Combination

Create combined loadcases from a Nastran OP2 file using linear superposition.

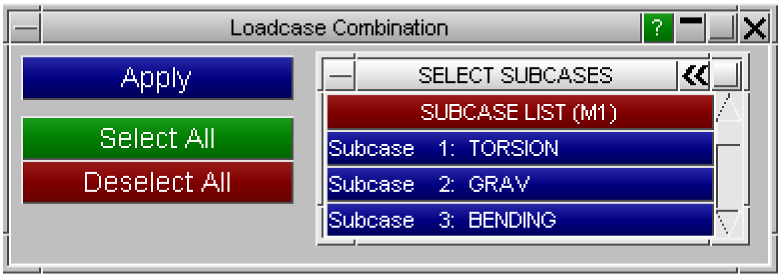


Loadcase Name

Specify a name for the combined loadcase.

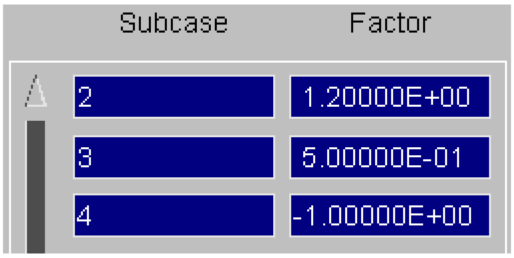
Select Subcases

Select the subcases to combine



Set Factors

Factors can be applied to the selected subcases (Default = 1.0)

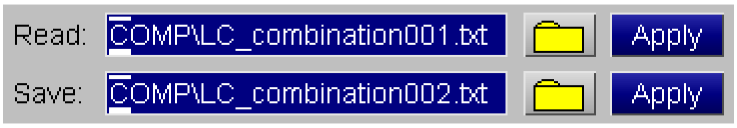


Create Combined Loadcase

Press **Apply** to create the combined loadcase

Read/Write Combinations

To save time having to create different loadcase combinations each time a new D3PLOT session is started, the combinations can be saved to a text file. The combinations can then be read back into D3PLOT in later sessions.



The file is a simple text file and could be generated by an external program as part of an automatic process. The format is:

START_SUBCASE,<name>

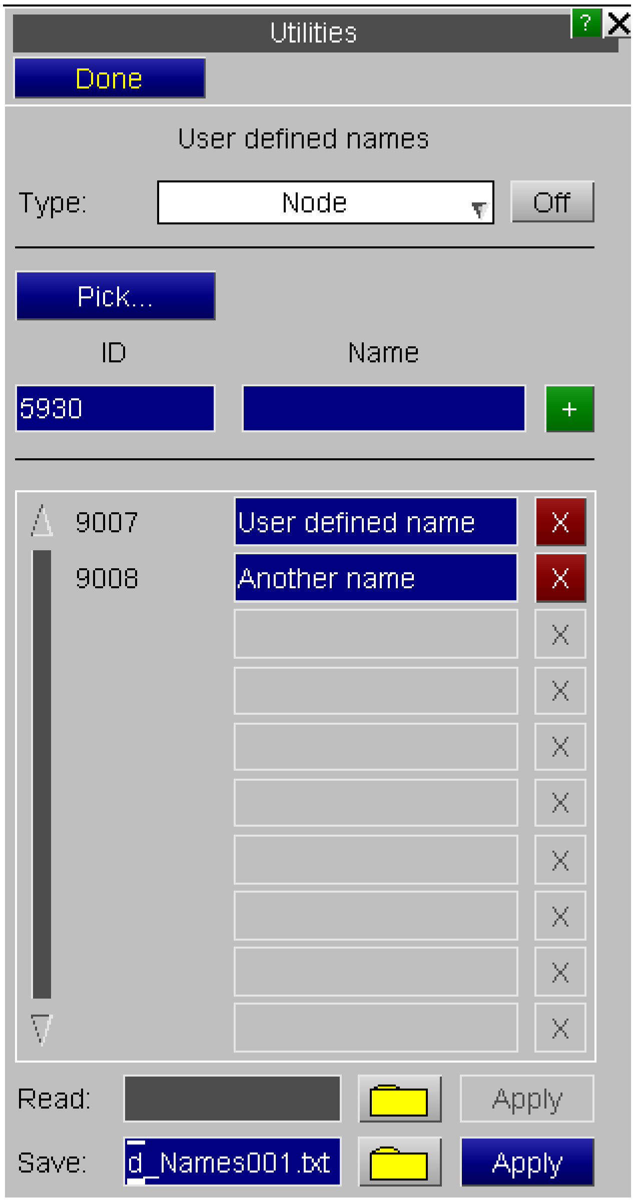
```
START_SUBCASE,<name>
<subcase_id>, <factor>
<subcase_id>, <factor>
<subcase_id>, <factor>
.
.
etc.
END_SUBCASE
START_SUBCASE,<name>
<subcase_id>, <factor>
<subcase_id>, <factor>
<subcase_id>, <factor>
.
.
etc.
END_SUBCASE
.
.
etc.
```

The file is not model specific so you could save combinations from one model and read them in for another. If a selected subcase ID doesn't exist in the new model it is ignored.

9.10.18. User Defined Names

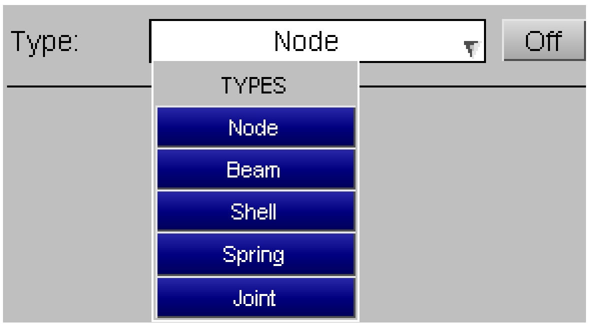
User Defined Names

Define names for entities to display on the model.



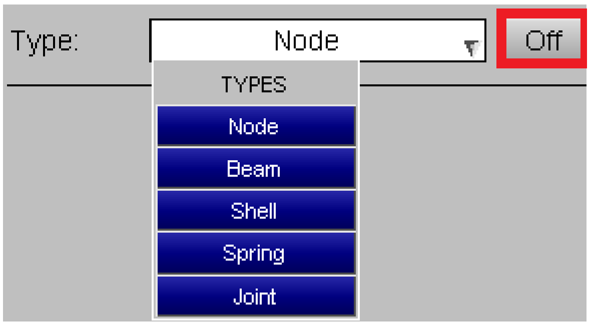
Select Entity Type

Specify an entity type to define names for. The list of available entities will depend on the contents of the model.



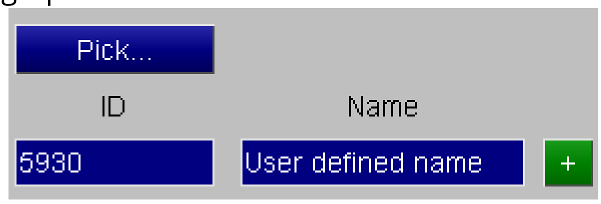
Toggle Display On/Off

Toggle the display of names.



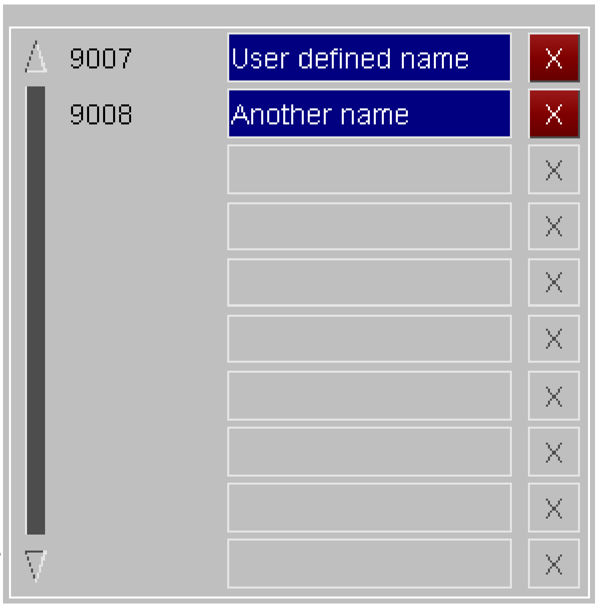
Pick Entity to Name

Press the Pick button to pick an entity in the graphics window and then define a name. Press the '+' button to add it to the list.



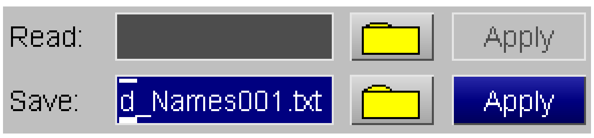
Edit/Delete Names

Previously defined names can be edited or deleted.



Read/Write Names

To save time having to define names each time a new D3PLOT session is started, the names can be saved to a text file. The names can then be read back into D3PLOT in later sessions.



The file is a simple text file and could be generated by an external program as part of an automatic process. The format is:

NODE_USER_NAME_START

```
NODE_USER_NAME_START
<node_id>, <name>
<node_id>, <name>
<node_id>, <name>
.
.
etc.
NODE_USER_NAME_END
SHELL_USER_NAME_START
<shell_id>, <name>
<shell_id>, <name>
<shell_id>, <name>
.
.
etc.
SHELL_USER_NAME_END
.
.
etc.
```

The file is not model specific so you could save names from one model and read them in for another. If an entity ID doesn't exist in the new model it is ignored.

9.10.19. GSA Data

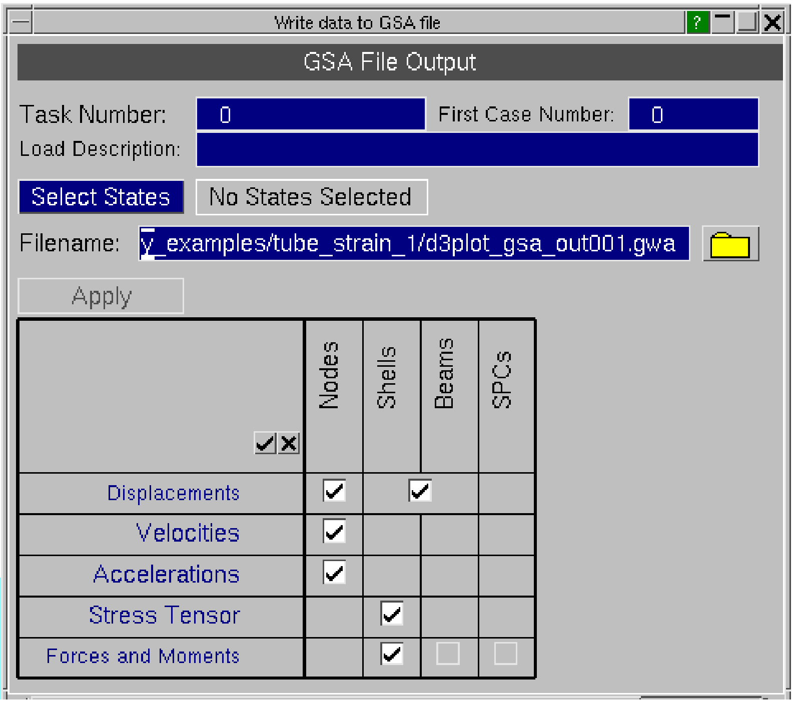
GSA Data

This is available from D3PLOT 13.0 and compatible with GSA 7.2.

This figure (right) shows the menu for writing an ASCII file of results in .gwa format that can be read into GSA for the corresponding analysis case.

Where multiple states are selected, each state will be output in order as a new analysis case with an incremented case number.

If a component is not available in the D3Plot model the option is greyed out. SPCs have not been fully implemented so their selection is always greyed out.



TASK NUMBER	Input the task used in the corresponding GSA analysis case.
FIRST CASE NUMBER	Input the first case number used in the corresponding GSA analysis case.
LOAD DESCRIPTION	Input the load description used in the corresponding GSA analysis case.
SELECT STATES	Select the states you want to output GSA data for.

Limitations:

1. D3Plot shell and beam data is only output as averaged centre values.

2. The forces and moments element data is in the local topological axis (rather than a user defined or global axis).

9.10.20. Streamlines

Streamlines

Streamlines are available for ICFD Volume data from D3PLOT 14.0.

Streamlines are created in groups, see Edit groups . Streamlines can be displayed differently and use different data components in different windows, see Streamline Window Settings . Settings for all streamlines can be changed in Advanced options . For more information on how the streamlines are computed see Computational Methods .

Streamlines are displayed when the graphics window is refreshed. The streamlines menu can be dismiss by clicking **DONE** .

Reason for a streamline end	Any further action
The streamline reaches the ICFD Volume boundary.	None
The streamline is in an area where the velocity magnitude is less than the **STAGNATION SPEED** (see Advanced options).	Message displayed in the dialogue box saying how many streamline(s) reach a stagnation point.
The streamline start point was outside the ICFD Volume domain.	Message displayed in the dialogue box the first time streamlines are computed for this state and streamline group saying how many streamline(s) are outside the domain.
The streamline computation is unable to identify where the next step is located.	Message displayed in the dialogue box saying how many streamline(s) could not be computed to the boundary. (This count does not include any streamlines in the above two categories.) It is recommended that the step size is reduced in the Advanced options .

9.10.20.1. Edit Groups

Edit Groups

The Edit groups tab allows users to edit the groups of streamlines.

 Select the first group

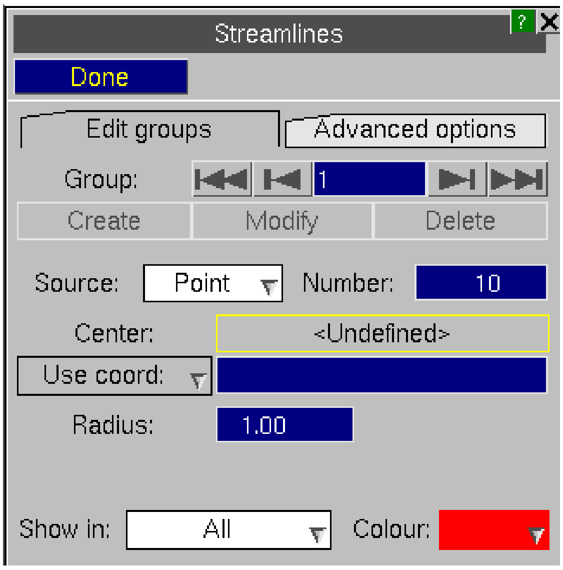
 Select the previous group

Go to this existing group

 Select the next group

 Select the highest group that exists + 1.

Users can **CREATE** new groups, and **MODIFY** or **DELETE** existing groups.



The start points for a streamline group can generated by a **POINT** source or a **LINE** source.

POINT Source	Start points are equally spaced on a the surface of a sphere with a given **RADIUS** and **CENTRE** . If the model is a 2D analysis, start points are equally spaced in a circle with a given **RADIUS** and **CENTRE** . (In 2D, the position of start points within the circle depends on the local axis of element 1.)
LINE Source	Start points are equally spaced on a line between **POINT 1** and **POINT 2** , both the end points are start points.

The centre, or point 1 and point 2, can be input in different ways:

Use COORD	Input a co-ordinate in model space.
Use ELEMENT	Input an ICFD Volume element ID in the format Di/j where "i" is the domain ID and j is the element ID.
Use NODE	Input an ICFD Volume node ID in the format Di/j where "i" is the domain ID and j is the node ID, or **PICK VISIBLE** to use a screen pick.

Options in the Edit groups menu are:

SOURCE	Select to use a POINT or LINE source.
NUMBER	The number of start points generated for this streamline group. Note, any start points which are not inside the ICFD volume are not displayed.
USE COORD / ELEMENT / NODE	Select how to input the Center or Point 1 and Point 2 values.

PICK VISIBLE	For **USE NODE** use a screen pick.
CANCEL PICK	For **USE NODE** cancel a screen pick.
RADIUS	Set the radius of a POINT source.
SHOW IN	The windows the streamline group start points are used in.
COLOUR	The colour of the streamline group, used for **Sh** aded, **Li** ne or **Hi** dden plots.

9.10.20.2. Streamline Window Settings

Streamline Window Settings

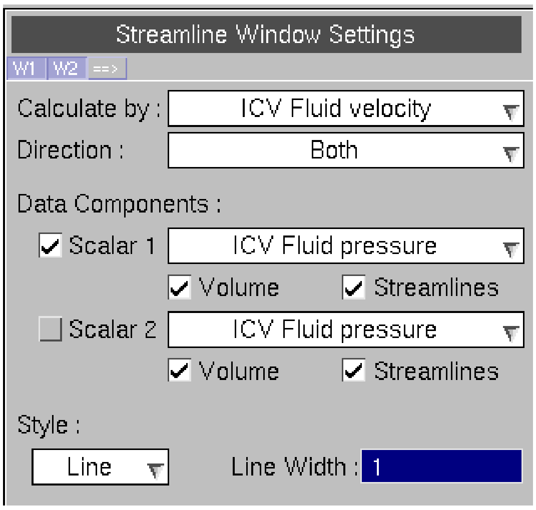
The Streamline Window Settings are located on the lower part of the streamlines menu. These are options that can be changed on a per window basis.

Note, the windows which streamline group start points are used in is set by the **SHOW IN** option on the Edit groups panel.

Streamlines can be calculated relative to the **ICV Fluid Velocity** vector field, or the **ICV Average Velocity** vector field (where this is available in the output data). The **ICV Average Velocity** is a time averaged velocity over the previous states.

Multiple data components can be used with great effect for streamlines, for more information on multiple data components see DATA COMPONENTS - ADVANCED .

The Streamline Window Settings panel gives users the option to change Scalar 1 and Scalar 2 to different ICFD volume components, to activate and deactivate Scalar 1 and Scalar 2, and select whether the data component is displayed on ICFD Volumes and/or on Streamlines.



CALCULATE BY	Select which velocity vector field is used to calculate streamlines.

DIRECTION	Streamlines can be displayed in BOTH directions from the start points, or only FORWARDS or BACKWARDS.
SCALAR 1	Activate or deactivate Scalar 1, and change Scalar 1 to a different ICFD volume component.
SCALAR 2	Activate or deactivate Scalar 2, and change Scalar 2 to a different ICFD volume component.
VOLUME	Select whether the above data component is displayed on ICFD Volumes.
STREAMLINES	Select whether the above data component is displayed on Streamlines.
STYLE	Streamlines can be displayed as a line or a tube (pending).
LINE WIDTH	The width of the streamlines.

Streamlines can be contoured with a scalar data component in **CT** , **SI** and **CL** plots. In other plotting modes the streamlines are rendered in grey.
Note, in **CL** oud plots the cloud points are positioned at each step of the streamline.

9.10.20.3. Advanced Options

Advanced Options

The Advanced options tab allows users to change the settings that are used for all streamlines. The image (right) shows the default values.

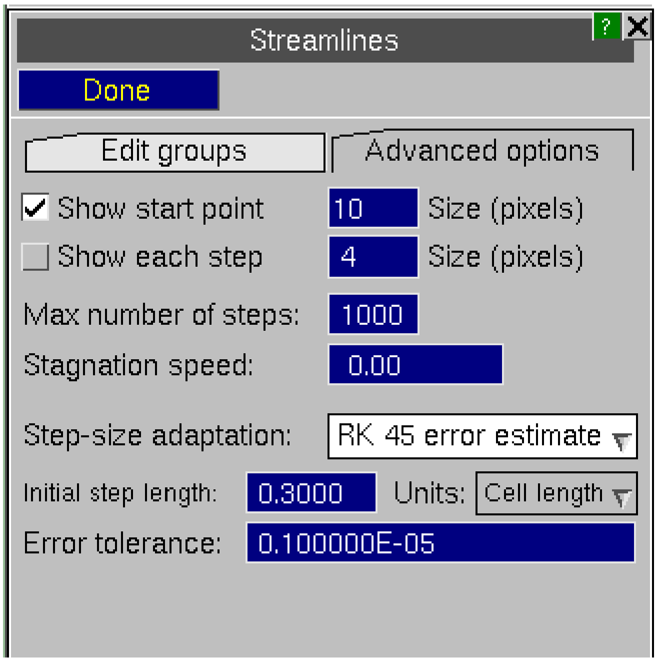
Each streamline is made of small line segments, called "steps". Ideally the steps should be short enough to give accurate results, but long enough to make it quick to compute the streamline. We use an adaptive step-size to get a balance between these.

The streamline can be lengthened (or shortened) by setting the **MAX NUMBER OF STEPS** to be used in the computation.

SHOW START POINT	Select whether the streamline start points are displayed and set their size.
SHOW EACH STEP	Select whether a point is shown at each step (small segment) of the streamline and set the point size. This can be helpful to check streamline quality.
MAX NUMBER OF STEPS	Change the maximum number of steps that can be used.
STAGNATION SPEED	Streamlines end (stagnate) when they reach an area where the velocity is less than the Stagnation speed .

There are four different **STEP-SIZE ADAPTATION** schemes available. Each scheme has parameters that can be changed. The relevant parameters are shown at the bottom of the Advanced options menu.

Use **HELP** in the **STEP-SIZE ADAPTATION** popup to see the following information about the different schemes.



Name	Description	Parameters	Recommended value (using cell length units)
RK 45 error estimate	This scheme is suitable for most applications. D3Plot increases the number of steps where the estimated error exceeds the **Error Tolerance** . If the start point is near an area of low velocity, you may need to reduce the **Initial Step Length** to improve accuracy at the the start.	**Initial step length**	0.3
		Error tolerance	1e-6
Target stepsize	This scheme gives equally spaced steps. The step length is the **Target step length** . The scheme becomes unstable if the velocity magnitude gets very small. To avoid this, either find as suitable **Stagnation Speed** at	**Target step length**	0.3

	which to stop the streamline, or use another adaptation scheme.		
Angle between steps	This scheme increases the number of steps where the streamline changes direction, and reduces the number of steps where the flow is parallel. Steps cannot be longer than the **Max step length** . If the streamline misses areas of interest, and to maintain accuracy where the flow is parallel you may need to reduce the **Max step length** . If the start point is near an area of low velocity, you may need to reduce the **Initial Step Length** to improve accuracy at the the start. This scheme can be used as a refined version of the **Target Stepsize** scheme by setting the **Max step length** to the **Target step length.**	**Initial step length**	0.3
		Max step length	3
RK 4 eigenvalue error	This scheme takes the longest time, but can give the most accurate and smooth results. D3Plot estimates a suitable step size such that the error for surrounding eigenvalues does not exceed the **Error Tolerance** . Steps cannot be longer than the **Max step length** . To maintain accuracy where the flow is parallel, or very different between adjacent elements you may need to reduce the **Max step length** . Note: For a 2D analysis this is only implemented where the 2D ICFD Volume is in the x-y plane (z = 0).	**Max step length**	3
		Error tolerance	1e-5

Step length parameters can be given in different **UNITS** :

CELL LENGTH	Values are relative to the length of the element the point is in. This allows step length parameters to vary where the model has a finer mesh.
LENGTH	Values are relative to units of length in the model.

For a more detailed description of step-size adaptation schemes see Computational Methods .

9.10.20.4. Computational Methods

Computational Methods

We compute each streamline using 4th order Runge Kutta methods. These methods compute the velocity at four nearby points and take a weighted average of them to get the velocity at the current point. The computation at each step is 4th order so *O(k^4)* accurate, where *k* is the (varying) step-size at that point.

Generally, decreasing the step-size increases the accuracy of the results, but also increases the time taken to compute a streamline. To get a balance between these, it is good to use an adaptive step-size. This means a suitable value of *k* is predicted before each step is made.

In simple terms, each step is given by *k.v* where *k* is the step-size and *v* is the velocity vector.

Typically, for 4th order Runge Kutta methods the maximum values of step-size *k* can be specified. This is useful as it directly corresponds to bounds on the accuracy of the computation *O(k^4)* .

In D3Plot we give maximum values of **step length** . This directly corresponds to the length of step *k.v* seen on the screen.

Example:
A *step-size* of 0.3 will give a step of length 3 where the velocity magnitude is 10, or a step of length 0.15 where the velocity magnitude is 0.5.
A *step length* of 1, will give a step of length 1 for any velocity magnitude.

Note, when the *velocity magnitude is near 0* , it can be difficult to calculate an appropriate step-size as *k = step length / velocity magnitude* .
In D3Plot, for the **RK 45 error estimate** and **Angle between steps** schemes if the start point is in an area of very low velocity, the first step for the streamline can be larger than desired; for the **Target Stepsize** scheme if the streamline enters an area of very low velocity this can cause the computation to become unstable (the streamline looks like a hedgehog).

The step length parameters can be given either in units of Length or Cell Length. Where Length is the units of length in the model, and Cell Length is the size of the element the streamline is in - this allows the step length to vary when models have a finer or coarser mesh.

The four different step-size adaptation schemes available in D3Plot are detailed below. Each scheme has parameters that can be changed. See the table in Advanced options for a brief overview of the schemes and their parameters.

RK 45 error estimate

Whilst conducting a 4th order Runge Kutta method, the error is also estimated in comparison to a 5th order Runge Kutta method.
This scheme is considered to be computationally efficient, but the error estimate is only *O(k^5)* accurate, which may mean the same **Error Tolerance** does not give as high a quality streamline as the **RK 4 eigenvalue error** scheme.

D3Plot increases the number of steps where the estimated error exceeds the **Error Tolerance** :
The step-size is *halved* if the *estimated error* > *Error Tolerance* .
The step-size is *doubled* if the *estimated error* < *10 x Error Tolerance* .

Note, accuracy is reduced if the start point is near an area of low velocity. Either reduce the **Initial step length** , or move the start point.

Target stepsize

D3Plot gives steps of a fixed length given by the **Target step length** .
At each stage *k* = *Target step length / velocity magnitude* .
Note, this scheme becomes unstable if the streamline enters an area of very low velocity. To avoid this, either find a suitable **Stagnation Speed** at which to stop the streamline, or use another adaptation scheme.

Angle between steps

D3Plot increases the number of steps where the streamline changes direction, and reduces them where the flow is parallel.
The *cos(step angle)* is given by the dot product of the current step and the previous step.
The step-size is *halved* if the the *step angle* > *15 degrees* .
The step-size is *doubled* if the *step angle* < *3 degree* s.

The scheme can loose accuracy if the steps are very big and there is a sudden change in velocity, to avoid this reduce **Max step length** .

RK 4 eigenvalue error

The *growth factor error* for the 4th order Runge Kutta method is estimated.
The *growth factor error* is the error for any *lambda.k* where *lambda* is a real or imaginary eigenvector (i.e. factor of growth) and *k* is a (varying) step-size.
D3Plot computes *LK_max,* the maximum value of *lambda.k* such that *growth factor error* < *Error Tolerance* .
To compute *LK_max* for each step it estimates the largest eigenvalue. This is done by computing the velocity at 3 infinitesimally nearby points. It is this aspect of the scheme which is time consuming.
A suitable *k* is chosen so at this time step *k.v* < *k.lambda* < *LK_max* .

Note: Generally, it is more efficient to use a bigger **Max step length** . However, if this is too large (over approximately 5 cell units), it takes longer for a scheme to locate which element the next point is contained in, than the time it saves in using fewer steps.

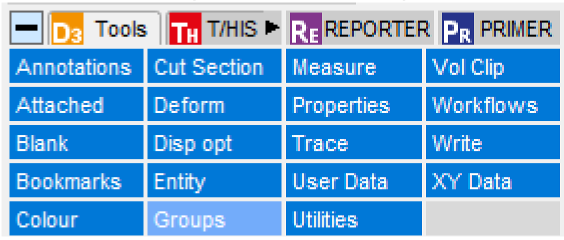
9.11. GROUPS

GROUPS:

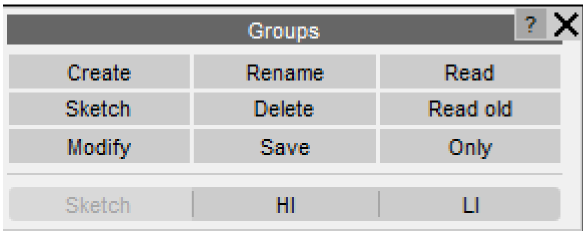
Groups as of D3PLOT 9.0 (Nov 2003) have been extensively rewritten to remove some limitations and to provide new features. In particular:

- You may now have an unlimited number of groups. The previous limit of 30 is removed.
- Groups are now automatically stored in a " `.grp` " file for this model, and are "remembered" across successive runs of D3PLOT, with no intervention required by the user. " `.grp` " files are model dependent binary files.
- The ascii *GROUP format used by PRIMER may be read in as " `.asc` " files, and converted to internal groups. These files may be created simply by hand, and are model independent. " `.asc` " files may also be written. They are also read by T/HIS, and this permits groups to be used consistently across the software suite.

Groups may be used in most contexts where selection takes place, for example **BLANK** , **WRITE** , etc.



They can be set up for a model in any of the following ways:



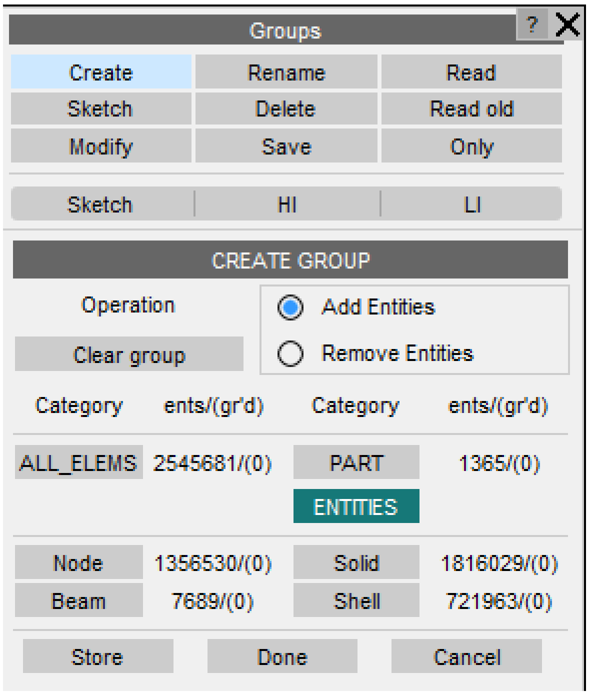
- Defined interactively during a D3PLOT run using **CREATE** .
- Read in from an external ascii (`.asc`) groups file
- Read in from an "old" binary (`.bin`) groups file
- Reread from a previous D3PLOT run, made available automatically via the < *jobname* > `.grp` file.

For backwards compatibility the "old style" binary groups files " `.bin` " may still be read (see READ_OLD... Reading "old" (pre version 9.0) binary .bin groups files), but they are no longer written and their use is deprecated.

9.11.1. CREATE

CREATE

Create groups by selecting an entity type (e.g. PART), then selecting using the normal methods. You must **STORE** the group to make it available for blanking and other operations. **STORE** saves the group to memory, but not to disk.



9.11.2. SAVE

SAVE

SAVE stores ***all*** selected groups in an ascii (.asc) groups file, making the groups portable across different models.

The format of this file is described in The format of the ASCII groups file (.asc file) below.

9.11.3. READ... Read Groups in from an Ascii Groups File

READ... Read groups in from an Ascii groups file

You can read in groups from any ascii groups file, including one not written from this model. The external data is converted to internal format and saved in the model's binary `.grp` file. (This is done for speed: reading and deconstructing an ascii groups file every time a group is used or edited would otherwise be very slow in a large model.) From D3PLOT 21.0, the `.grp` file stores the visual attributes of groups, if defined in the Ascii file.

The following rules apply:

Matching of external (file) and internal (model) data

- Matching of data between model and group file is via external labels.
- If an item exists in the model, but is not defined in the file, it will not be grouped.
- If an item exists in the file, but not in the model, it will be ignored - this is not an error.

Groups label and numbering policy

- Groups in an ascii file have unique labels <#i> <#j> ....
- They will be transferred to become the same internal group labels in the model's `.grp` file.
- If group <#n> already exists in the model's `.grp` file ***it is overwritten - unconditionally and with no warning.***

Therefore you may read in any number of ascii group files, and the most recent definition of a group in such a file will be the current one.

Automatic reading of ascii groups files

Two types of default ascii groups files may optionally be read in automatically when a new model is opened:

• "Master" group file	This may be defined in the "oa_pref" file (see Appendix B) using the **`d3plot*master_group_file: < filename >`** option. This will be read in every time a new model is opened, and the definitions in it will be mapped onto that model's groups, creating or replacing groups as required.
• "Local" group file in current directory	If the " **`Read ASC file (Ascii groups file)`** " box on the model reading panel (see Open a Single Model) is selected then the most recent file: **`d3plot_< nnn >.asc`** (eg **`d3plot_003.asc`**) in the directory where the model is found will be read in, replacing or creating groups as above. Note "most recent" may not necessarily be the file with the highest < *nnn* > number - it will depend on the file's creation date.

These two files are read in the order above, thus a definition in a "local" groups file can supersede one in the "master" one.

Reading of these files when a model is opened can be suppressed with the command line " **`-no_group`** " option, also with the " **`d3plot*read_asc_file: true |false`** " preference.

9.11.4. The Binary jobname.grp File

The binary jobname.grp file

When a group is read from an ascii file, or created interactively in a D3PLOT session, the information is stored in a binary " *jobname* .grp" file. There are two reasons for this:

1. D3PLOT permits an unlimited number of groups, so storing this information in memory could become expensive. Storing each group in an ascii file would be possible, but the processes of agglomerating data to write this file, and then deconstructing it prior to re-reading and use are both time-consuming and this would make the editing and use of groups unacceptably slow. The binary ".grp" file contains group information organised in a model-specific way that is compact and very fast to read and write, making it a much better solution for storage and editing.

2. You are not required to write ascii groups files when a D3PLOT session terminates, so keeping the binry ".grp" file means that group information is remembered if the model is subsequently reopened in a later D3PLOT session.

What happens to stored groups in the jobname.grp file when an ascii groups file is read?

When an ascii groups file (group *nnn* .asc) is read, either manually or automatically when a model is opened, its contents are read and stored. If it contains a group id that is currently defined in a binary jobname.grp file then the contents of the ascii file will overwrite (ie supersede) the contents of the binary file, becoming the current definition of group #id. Any new group ids not in the binary file are added to it, but existing groups in the binary file not present in the ascii one are not deleted.

The "-no_group" command-line option does not influence the reading of binary jobname.grp files.

The " **`-no_group`** " command-line option (and the " **`read_asc_file`** " preference) do not affect the reading of any existing binary *jobname* .grp file when a model is opened. So these options can be used to prevent ascii group file definitions superseding groups remembered from earlier D3PLOT sessions using this model.

9.11.5. READ_OLD... Reading "Old" (Pre-D3PLOT 9.0) Binary .bin Groups Files

READ_OLD... Reading "old" (pre D3PLOT 9.0) binary .bin groups files

For backwards compatibility the ability to read older `.bin` binary groups files has been preserved, however D3PLOT can no longer write these files.

These files contained up to 30 groups which, in pre-D3PLOT 9.0 versions, were hard-wired as labels 1 to 30. When read back in the following rules are applied:

Matching of internal and external data

- The same rules that are used for ascii groups are applied to binary ones.
- Items in the `.bin` file, but not in the model, are ignored.
- Items in the model, but not in the `.bin` file, are not grouped.

Group labelling

- Each group in the `.bin` file is stored as the first free group id for this model.
- Therefore assigned group ids will depend on what has been created &/or read in previously.
- You are informed about the label assigned to each group as it is read in.

The use of "old style" binary (.bin) groups files is deprecated

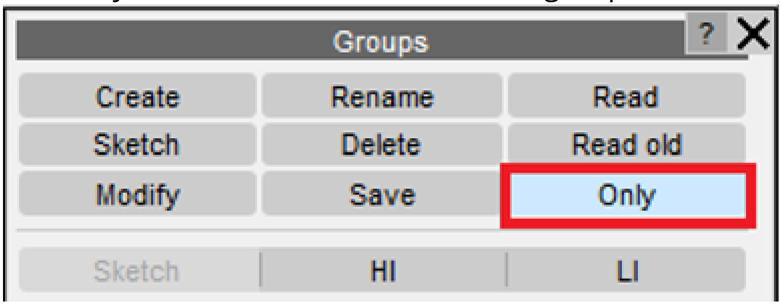
Binary groups files can be written by earlier versions of D3PLOT, and also by PRIMER. However their use is discouraged, and the much more compact and easily edited ascii groups files are recommended instead. This is because:

- Ascii groups files tend to be much more compact.
- They are easily created and modified by humans using a normal text editor.
- They can be read by PRIMER as part of a keyword deck (the special *GROUP keyword after *END)
- They can be read by T/HIS.
- And as ASCII format files there are no issues of binary compatibility across different machines.

9.11.6. ONLY

ONLY

ONLY in the Groups panel unblanks only the entities from the selected groups and blanks all the other groups.



Tools >> Groups >> Only will give a list of all the groups. Selecting any group from the list will display only the selected group entities in the graphics area with the visual properties (colour, transparency etc) defined for that group.

Performing an ONLY operation on groups from Part Tree also restores the visual properties of the selected group.

This feature of restoring the visual attributes of groups with an ONLY operation can be widely used while working with the Human Body Models, to have a clear visualisation of HBM entities. From D3PLOT 21.0, the HBM visualisation entities can be loaded into D3PLOT by reading in a "*.vis" file (which is created from the HBM visualisation table in PRIMER) from the "Groups >> Read" panel.

9.11.7. Read HBM Visualisation entities

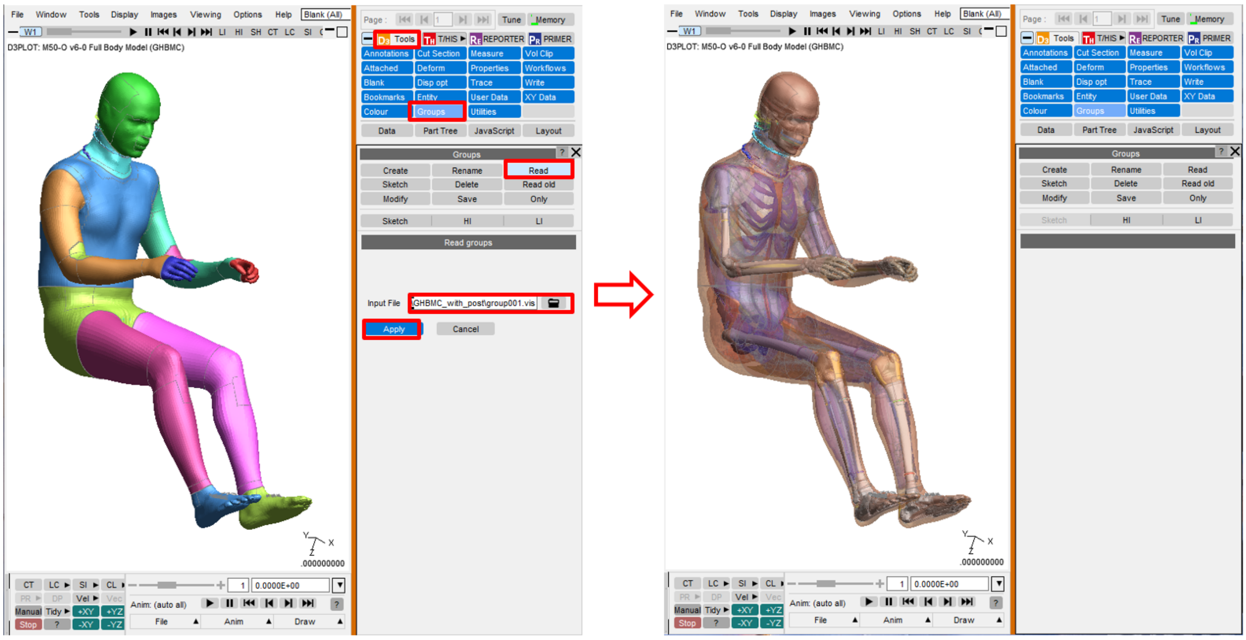
Read HBM visualisation entities (*.vis file)

It is possible to view the HBM visualisation entities in D3PLOT by reading in the HBM visual entities related groups from a ***.vis** file, created from HBM visualisation table in PRIMER.

The following rules apply for ***.vis** file:

1. It should be loaded with the LS-DYNA results (*.ptf/d3plot file) of the same model for which it is created in PRIMER.
2. This file is not read in automatically when a new model is opened for the first time in D3PLOT.
3. This file contains the visual properties details (colour, transparency, blanking status etc.) of the HBM visualisation related groups.
4. The visual attributes of the groups are applied on the model in the graphics area as soon as the groups file is read.

The following image shows the steps to read the *.vis file. (**Model acknowledgements: GHBMC - Elemance**)

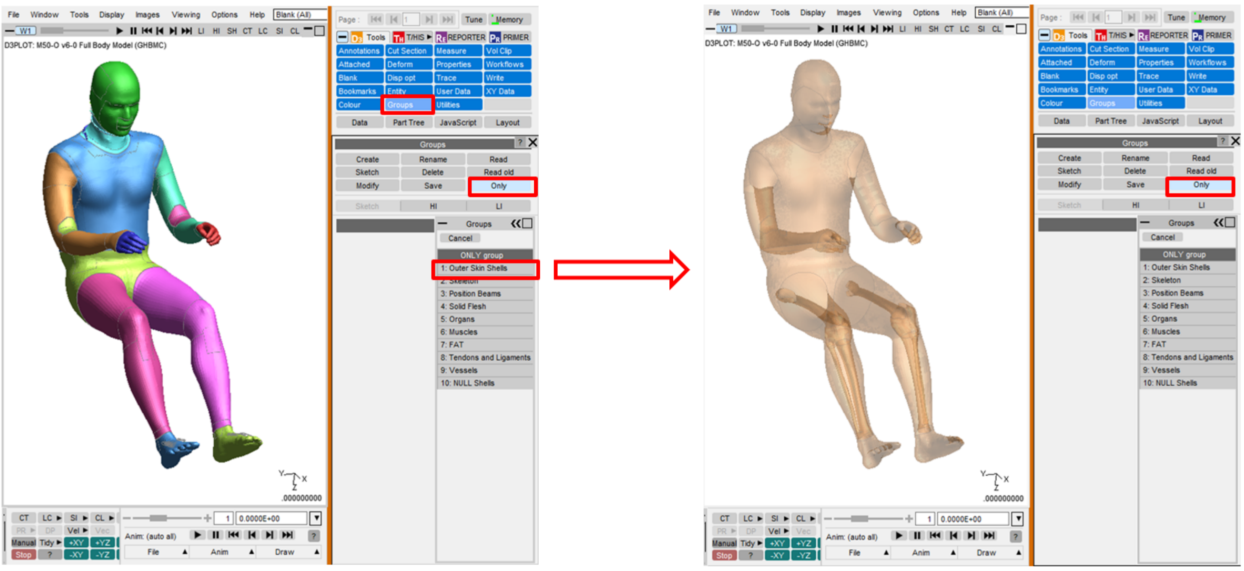


The following images show the workflow to view only a particular visual entity of an HBM. This can be done via the **Groups panel** or the **Part Tree**.

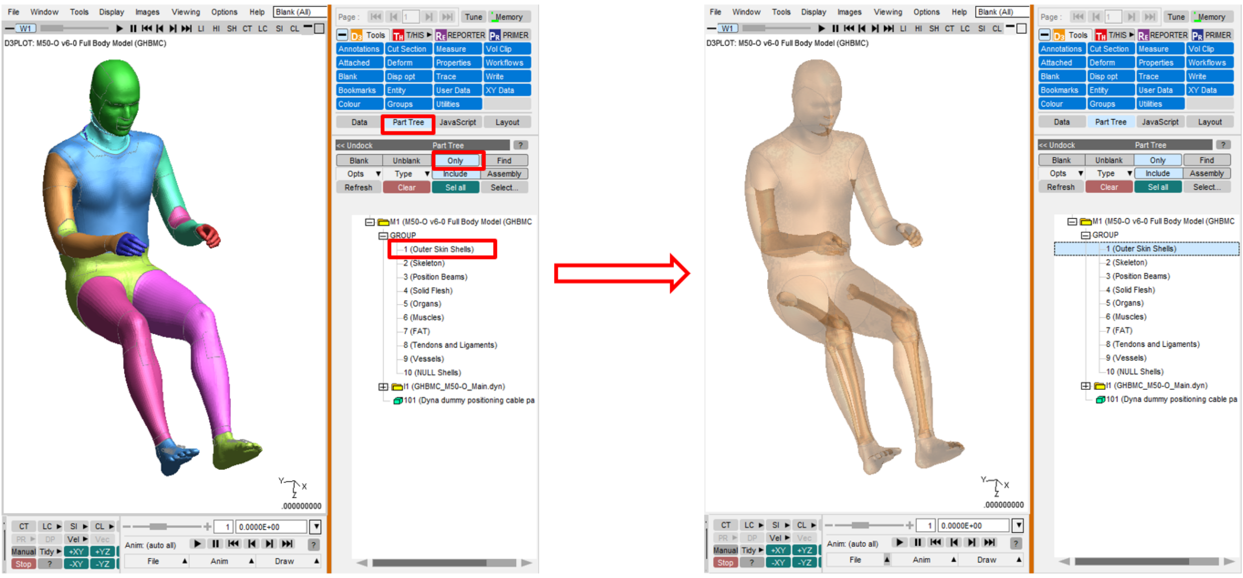
In the example shown in the images, the visual properties are defined in a *GROUP for the "Outer Skin Shell" entities in a Human Body Model (Colour - Orange, Transparency - 70%).

These entities in the "Outer Skin Shells" group are displayed with the group visual properties, when the group is selected under ONLY menu in the docked panel.

HBM Visualisation with Groups → Only



HBM Visualisation with Part Tree → Groups → Only



9.11.8. The Format of the ASCII Groups File (.asc File)

The format of the ASCII groups file (.asc file)

A file contains one or more `*GROUP` definitions, formatted as follows. All entry is free format.

<table>
<tr><td>

```
*GROUP
< title >
< label >
(<props>)
```

</td><td>This is a header designating a new group.
<title> may be up to 80 characters wide
<label> is in the range 1 to 99999999 (However large numbers are deprecated)
<props> are optional graphical properties to be applied to this group - see < Properties > below.</td></tr>
<tr><td>< Properties ></td><td>From D3PLOT 9.3.1 onwards, the following optional "properties" may be added, in free format (space or comma separated) in columns 11 to 80 following < label >
<table>
<tr><td><colour></td><td>colour name, or RrrrGgggBbbb</td></tr>
<tr><td><transparency></td><td>0 - 100</td></tr>
<tr><td><display mode></td><td>WIRE, HIDDEN, SHADED, CURRENT</td></tr>
<tr><td><overlay mode></td><td>NONE, FREE, ALL, CURRENT</td></tr>
<tr><td><overlay colour></td><td>colour name, or RrrrGgggBbbb</td></tr>
<tr><td><brightness></td><td>0 - 100</td></tr>
<tr><td><shininess></td><td>0 - 100</td></tr>
<tr><td><blanking status></td><td>BLANKED / UNBLANKED</td></tr>
</table>
Only required fields need to be defined, and trailing fields that are omitted result in "no change" to the relevant property.
If intermediate fields are not to be changed then an asterisk "*" should be inserted. For example the line:
<pre>10 red * * green * 70</pre>
Means:

<group label 10>
<colour contents red>
<no change to display mode>
<no change to overlay mode>
<overlay colour green>
<no change to brightness>
<shininess 70%>
<no change to blanking status

This "property" information is applied when the group is read in from ASCII file, but in D3PLOT 9.3.1 was not stored inside D3PLOT. If the group was exported to file again this information was not present. From D3PLOT 9.4 onwards, if the group is exported this information is now present, updating properites if they have
</td></tr>
</table>

<table>
<tr><td></td><td>changed. There are a few rules that D3PLOT follows when writing the "property" information:

• If the entities in a group all have the same value for a property then this is what is written;
• If the entities in a group have different values for a property, and the "Use first entity properites" switch is on, the value of the property of the first entity in the group is written;
• If the entities in a group have different values for a property, and the "Use first entity properites" switch is off, a "*". is written.

PRIMER neither reads, stores nor exports this information.

This "properties" line is an interim development inserted for D3PLOT 9.3.1 and this capability will be developed in the future.</td></tr>
<tr><td><item> <range>
<item> <range>
<item> <range>
etc</td><td>
<table>
<tr><td>< item > must be one of</td><td colspan="3"><code>PART, SURFACE, NODE, SOLID, SHELL, BEAM, TSHELL, MASS, DISCRETE, SEATBELT, ACCELEROMETER, SLIPRING, PRETENSIONER, JOINT, RIGIDWALL, SEGMENT.</code>

If preceded by a minus sign (eg <code>-PART</code>) then these items are removed from the group.</td></tr>
<tr><td rowspan="3">< range > must be one of:</td><td><code>ALL</code></td><td>all items in that category</td><td>eg <code>PART ALL</code></td></tr>
<tr><td><code><i> : <j></code></td><td><start> to <end> range</td><td>eg <code>SHELL 10 : 200</code></td></tr>
<tr><td><code><i j k l m></code></td><td>Up to 5 discrete labels on a line</td><td>eg <code>SOLID 10 20 33 45 200</code></td></tr>
</table>

As many lines as required to define the group may be used.</td></tr>
<tr><td>BOX suffix is Unsupported</td><td>The "proper" group definition in PRIMER also supports the suffix <code>BOX</code> <code>< label ></code> at the end of a line, meaning that the preceding definitions on that line are limited to what lies within the *BOX definition.

Because D3PLOT doesn't "know" about boxes these are ignored, and you should avoid the <code>BOX</code> suffix if you are planning to use group files outside PRIMER.</td></tr>
</table>

***END**	This is optional, and is taken to mean the end of the file. A physical <end of file> is treated as terminating the last group in the file, and no ***END** is required.

Any number of group definitions may exist in a file, with the next ***GROUP** header effectively terminating the previous group definition.

The easiest way to create a groups file is to write one from D3PLOT, and then look at it in a text editor. The format is extremely simple and easy to understand.

Warnings:

a) D3PLOT groups contact surfaces, if they are present, by interface segment.

Since segments are not numbered in a LS-DYNA input deck, or indeed may not even be present for contacts defined by anything other than segment sets, attempting to read these back into PRIMER will fail and such definitions should be edited from ascii groups files if they are to be used for this purpose.

b) Group files written by PRIMER may contain **BOX** arguments, limiting the geometric region in which items are included.

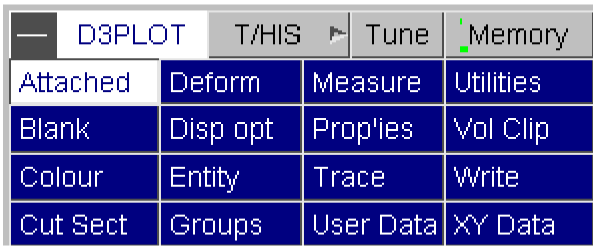
Because D3PLOT doesn't "know" about boxes it cannot apply these, and they are ignored. Therefore if you are intending to create groups that are portable across programmes it is recommended that you do not use the **BOX** argument.

9.12. ATTACHED

ATTACHED

The **ATTACHED** menu can be used to find entities that are physically attached together.

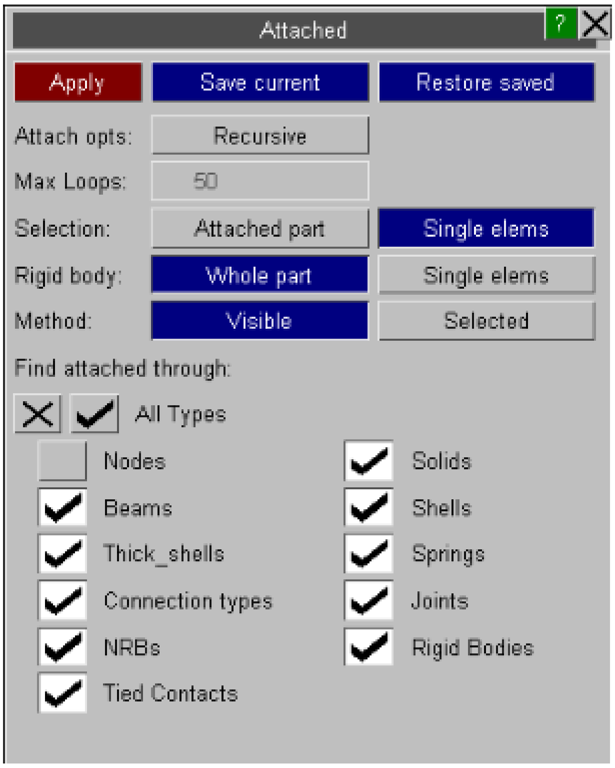
Each time you press **ATTACHED** D3PLOT does the following:



- Looks at what you want to find attached (beams, shells etc.).
- Find what is immediately "attached to" what is currently visible .
- Unblanks these newly found items
- Redraws the image.

The result is progressively more and more of the model being drawn until nothing attached to what is currently visible (which is not necessarily the whole model) remains to be unblanked and drawn. There is a pre-programmed shortcut for pressing **ATTACHED** that can be assigned to a shortcut key if required.

NOTE: This is slightly simpler than the Attached function in PRIMER as it only finds items attached at nodes.



9.12.1. Attached Options

Attached options

There are a few options available to the user to increase the flexibility of the attached panel.

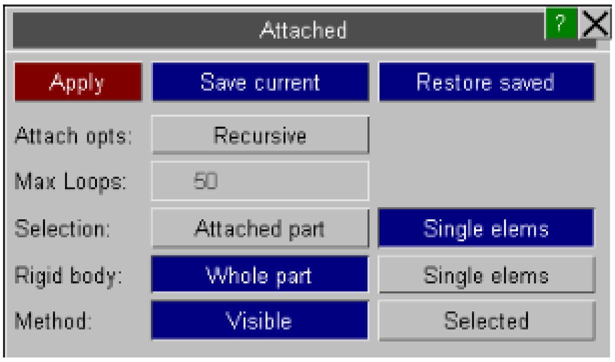
On entering the panel the current blanking status of the model is saved and can be restored with the **RESTORE SAVED** button. The blanking status can be saved at any point by pressing the **SAVE CURRENT** button. This can be useful if too many entities are revealed, making it possible to repeat the process with some attached categories switched off (see Restricting the extent of "attached to" propagation).

RECURSIVE will iteratively keep finding attached items until no more can be found or **MAX LOOPS** is reached. (The STOP button can also be pressed if it is taking too long).

Setting Selection to **SINGLE ELEMS** on means each time Apply is pressed only the entities immediately attached will be made visible. Setting Selection to **ATTACHED PART** on will make the parts attached visible.

Setting Rigid body to **WHOLE PART** on or **SINGLE ELEMS** on is analagous to setting Selection to **ATTACHED PART** on and **SINGLE ELEMS** on respectively but only applies to rigid parts. This option is only available if a ZTF file has been read in. If **WHOLE PART** is on then non-contiguous rigid parts will be found.

Instead of finding attached to all visible nodes, the user can select the nodes to find attached to. This is done by selecting **SELECTED** for the method instead of **VISIBLE** .

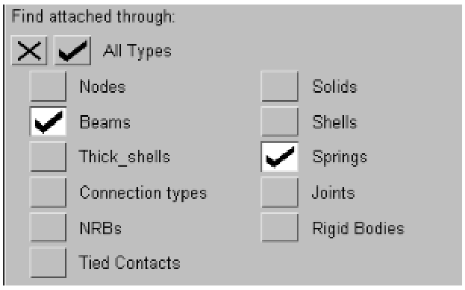


9.12.2. Restricting the Extent of "Attached to" Propagation

Restricting the extent of "attached to" propagation

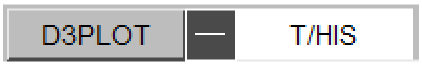
It is possible to limit what is found attached through entity switches.

In the example opposite, you could still display shells, solids, beams etc. but just find attached beams and springs. (Note that Constrained Rigid Bodies and Tied Contacts cannot be displayed but attached entities can be found through these types.)



9.13. T/HIS the D3PLOT <=> T/HIS Link

T/HIS the D3PLOT <=> T/HIS link



The **T/HIS** command starts T/HIS in background and establishes a link between the two codes, allowing D3PLOT to access and display "time-history" data, and to synchronise time-history display in T/HIS with the current state displayed in D3PLOT.

Linked T/HIS has the following attributes

- One or more fully functional T/HIS graphics windows running within the D3PLOT user interface.
- The two codes "talk" to one another and share information about time-history data.
- Time-history "timeline" plots can be synchronised to show the current state in D3PLOT
- Accessing a particular time in either code will cause the other to update to show the same time
- Plotable items in time-history plots are displayed in D3PLOT, and may be screen-picked for selection in T/HIS.

When the T/HIS link is active the user interface operates either in D3PLOT native mode (as in the rest of this manual), or T/HIS native mode. You need to be in the correct mode to access commands for the relevant program, and you swap between modes using the **D3PLOT** and **T/HIS** buttons on the top right.

9.13.1. The T/HIS Panel

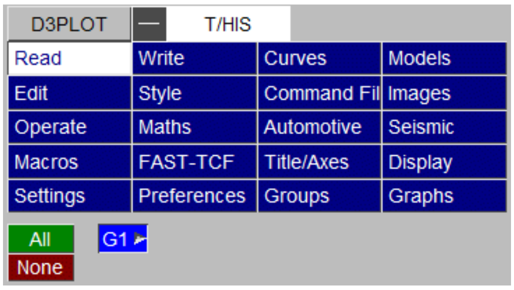
The **T/HIS** panel

When you first click on **T/HIS** the following happens:

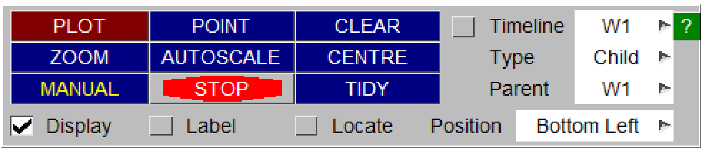
- D3PLOT silently opens a background T/HIS process
- This is passed the name of the current model.
- It searches for files called < ***name*** > that are readable by T/HIS.

- If found these are scanned and read into T/HIS

As shown here the top level T/HIS panel is identical to the "native" T/HIS top level commands menu, with the addition of some options specific to the linked case.



In addition to the top level panel changing to display the T/HIS commands the D3PLOT viewing menu is replaced with the T/HIS Global Command menu.



If the top level panel is changed back to display the D3PLOT commands the D3PLOT viewing panel will be redisplayed.

9.13.1.1. Association Between Models in D3PLOT and T/HIS

Association between models in D3PLOT and T/HIS

When T/HIS is started it will scan for time-history information for all models currently open in D3PLOT, and will continue to maintain associativity between the two codes. For example model M2 in D3PLOT will also be model M2 in T/HIS.

As models are opened and closed in D3PLOT so T/HIS will also open and close them, maintaining parity of model numbering. Internal flagging within D3PLOT of items being available for time-history processing will also be maintained.

WARNING: Opening a model in linked T/HIS only, rather than via D3PLOT, breaks the associativity between the two codes .

It is not illegal to open a model directly in linked T/HIS, but it has side-effects:

- Because D3PLOT cannot "know" about such a model's attributes it cannot be processed in D3PLOT in any way
- To make this clear models opened in this way are given model numbers 101, 102, ... in T/HIS

9.13.1.2. Use of T/HIS Derived Time History Data in D3PLOT

Use of T/HIS derived time-history data in D3PLOT

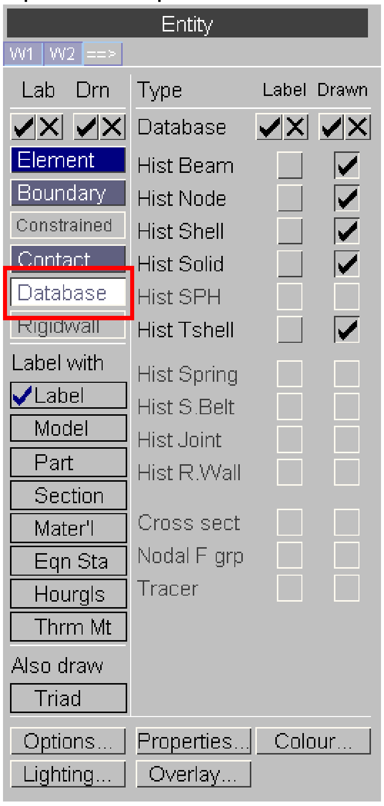
D3PLOT can display several items (for example springs, joints, elements and nodes in time-history blocks) that can have "time-history" information associated with them. When the T/HIS link is active this is handled as follows:

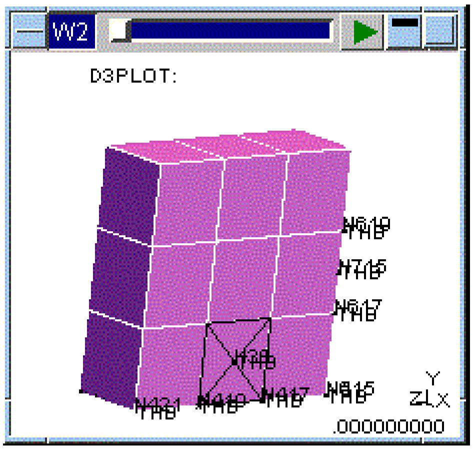
- Items in time-history blocks (ie those with **`*DATABASE_HISTORY_xxx`** cards) may be displayed by using the **Entity, Database** panel as shown here.

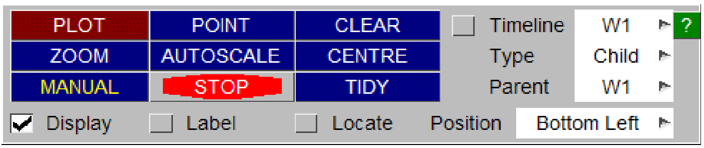
  In this example there are beams, nodes, shells, solids and thick shells in time-history blocks; but while two models are current only one of them contains time-history data, which is why the tick boxes have a grey background.

- Time history block symbols are displayed on elements as crossed lines, and on nodes as small squares. The image below shows the display of a solid and a few nodes in time history blocks (with labels turned on).

- Items not explicitly on *database_history cards, but which have time-history data available (springs, seatbelt elements, joints, rigid walls) may also have their visibility switched in this panel as "pseudo database history" items.





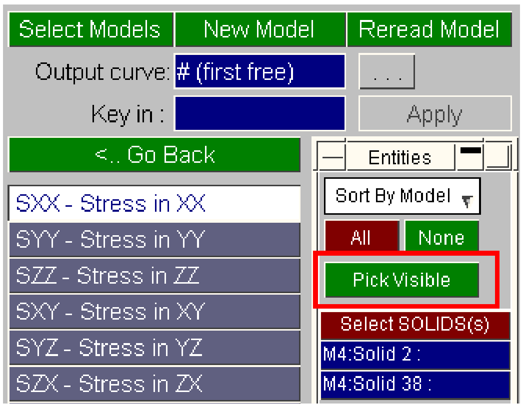


The display and labeling of items in "time-history" blocks can also be controlled using the **Mark Items** and **Label Items** buttons in the T/HIS Global Command menu.

Screen-picking items in Time-history blocks for selection in T/HIS

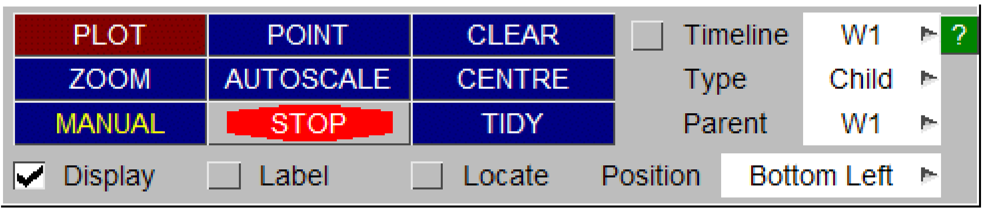
When items in time-history blocks are displayed in D3PLOT they can be screen-picked for selection in T/HIS menus by using **Pick Visible** .

Screen picking, menu selection and **Key in** can all be combined at will for the purposes of item selection.



9.13.2. Linked T/HIS Commands

Linked T/HIS commands



9.13.2.1. TIMELINE

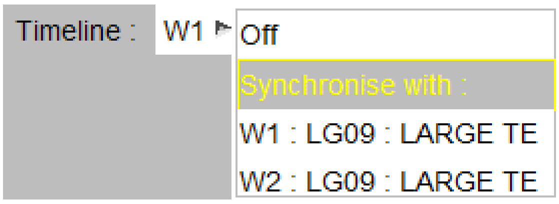
TIMELINE

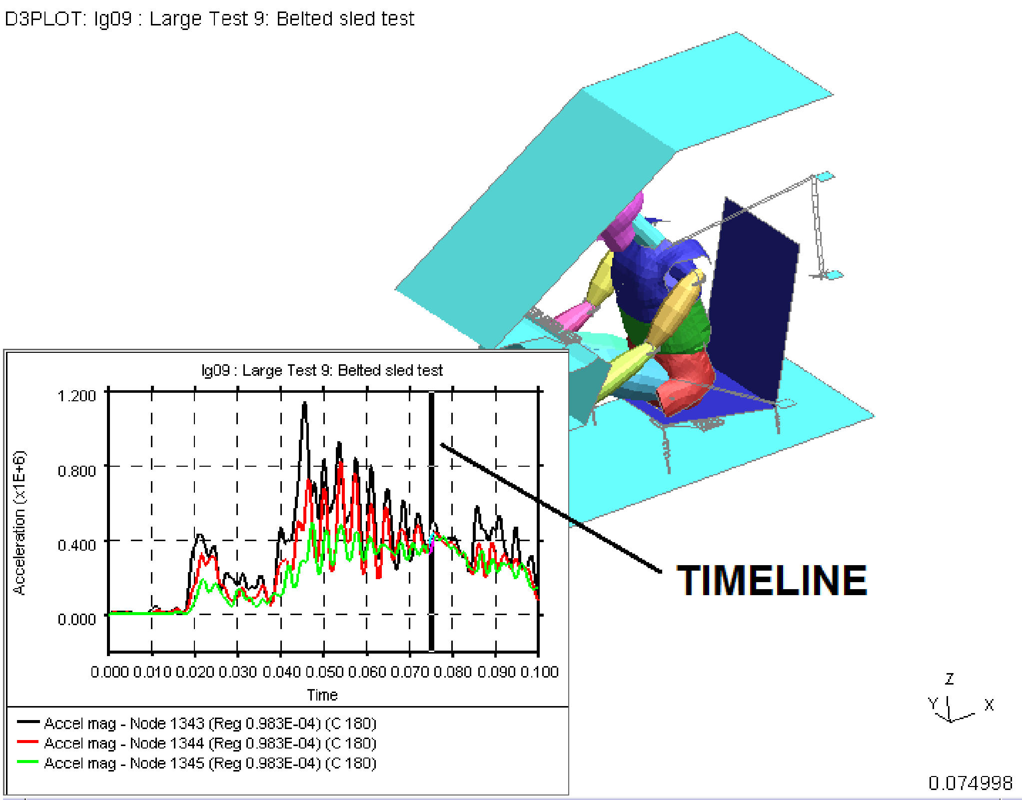
TIMELINE adds a vertical bar to the T/HIS plot showing the current time displayed in the D3PLOT graphics window. Each T/HIS graph can have it's own timeline.

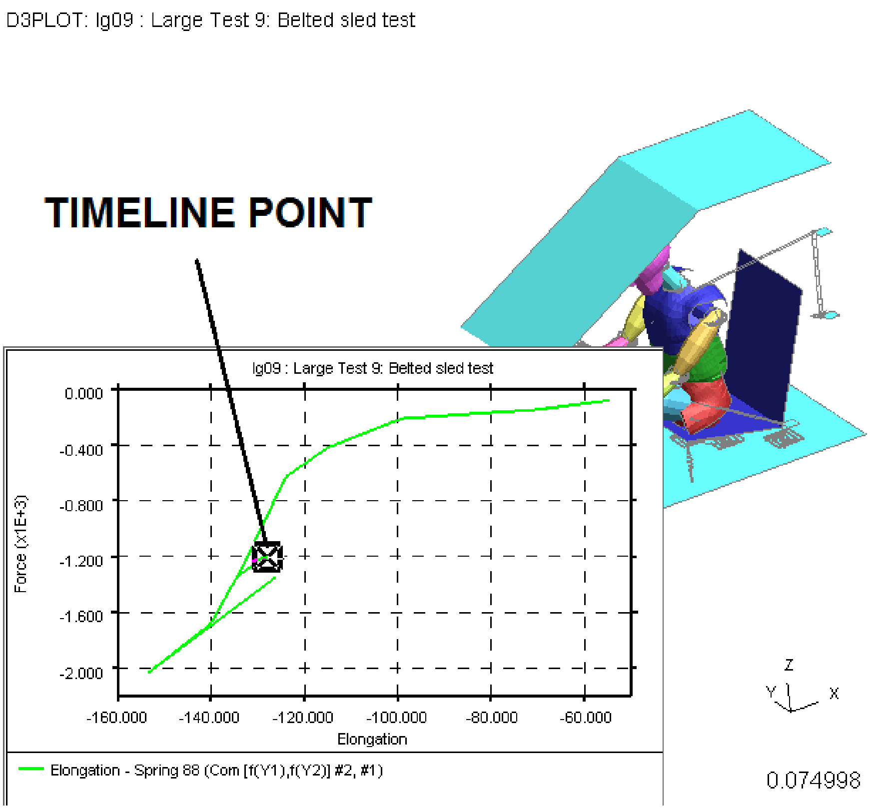
If a curve has been created using the T/HIS COM operation to combine two curves, e.g. Force-Displacement of a spring, the X-axis will no longer be time, so a vertical timeline makes no sense in this case. For curves that have been created this way, a point is drawn on the curve instead.

By default the first D3PLOT window will control the position of the timeline in each T/HIS graph.

The D3PLOT window used to control the timeline position can be changed using the popup menu in the T/HIS Global Command menu. This popup can also be used to turn **OFF** the timeline. This popup menu will set the timeline for ALL the T/HIS graphs.







After turning on the timeline in a T/HIS graph

- You can drag this bar in the T/HIS window to a new point, and the D3PLOT graphics window will jump to the new time.
- Equally if you move the D3PLOT graphics window to a new state the bar in the T/HIS plot will update.

Note that the timeline bar can only be positioned at times in the complete state (.ptf) file, since only these times are available as plotting states for D3PLOT. (Although D3PLOT can interpolate between states it would be potentially misleading to associate a "real" time-history value at some intermediate time with an interpolated graphical state.)

9.13.2.2. Graph Type

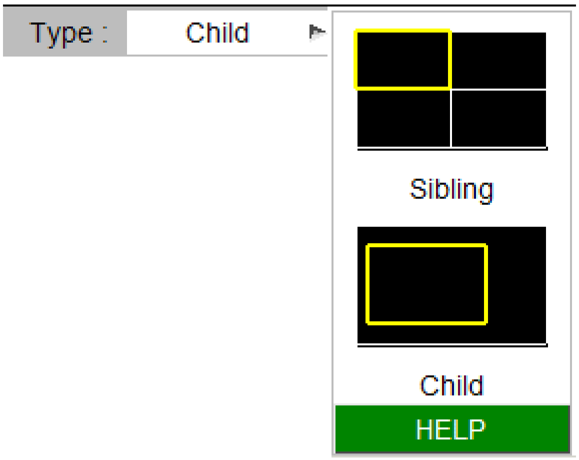
Graph Type

The T/HIS graph windows can co-exists with other D3PLOT graphics windows or they can be located within D3PLOT windows.

The type of each T/HIS window can be set to either

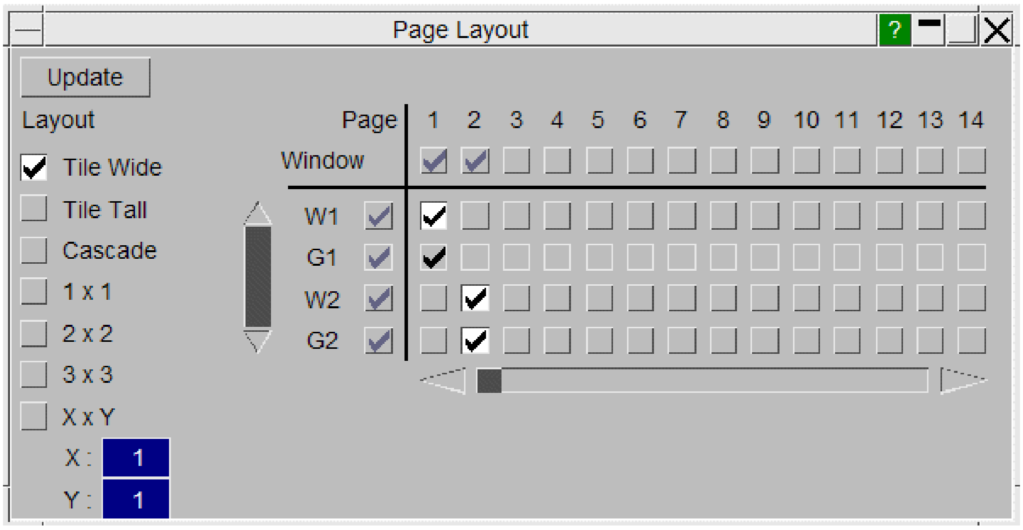
SIBLING	The T/HIS window occupies a "slot" in the graphics window layout, just like the other D3PLOT windows **`W1 .. Wn`**

CHILD	Makes it a child of a graphics window. If there is more than one you must choose which is to be its parent. It can be resized by dragging its borders, but may also have a preset **Window Position** in one of the four quadrants.



SIBLING T/HIS graphs are positioned on Pages using the same rules as other D3PLOT graphics windows (see Layout). **CHILD** T/HIS graphs do not take up a slot on pages.

The example opposite shows 2 D3PLOT windows (**W1** and **W2**) and 2 T/HIS graphs (**G1** and **G2**). Page 1 contains **W1** and **G1** (**G1** is a child of **W1**) and Page 2 contains **W2** and **G2** .



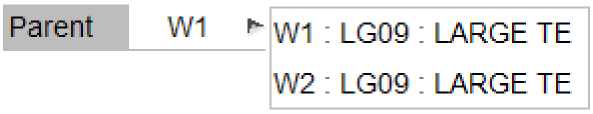
CHILD T/HIS graphs can have their window borders removed by docking the windows using the **Dock** option in the Toolbar of each window.

9.13.2.3. Parent

Parent

This option only applies to **CHILD** windows.

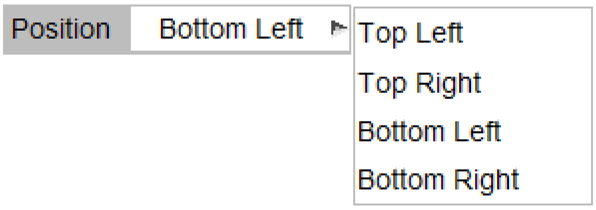
Each T/HIS graph can be positioned within any D3PLOT window.



9.13.2.4. Position

Position

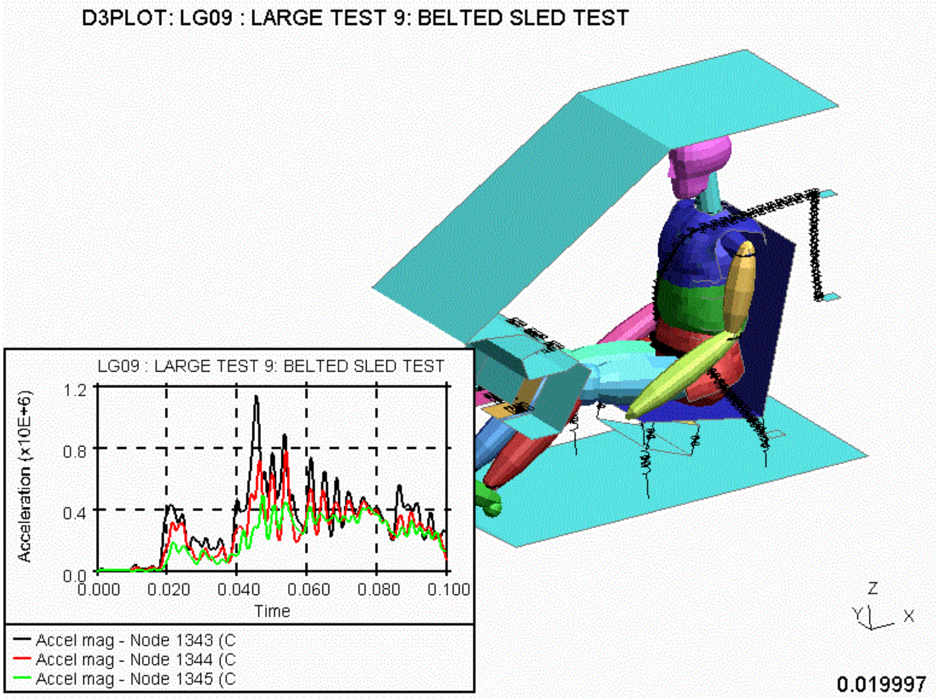
This option only applies to **CHILD** windows.



Each T/HIS graph can be positioned automatically into one of the four quadrants: bottom/top left/right.

Alternatively each T/HIS window can be moved and resized by clicking and dragging on it's borders and positioned anywhere within the parent D3PLOT window.

This example shows a **DOCKED** window located in the bottom right hand quadrant of the graphics window. No button bar is drawn because the mouse is not in that window.

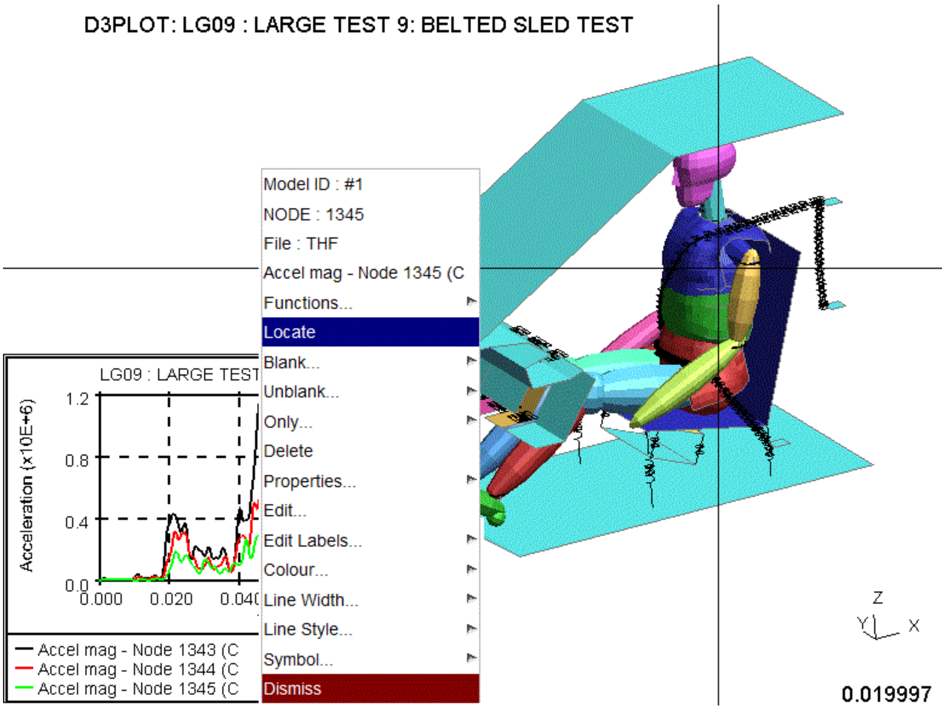


9.13.2.5. LOCATE Identifying a Curve's Item Visually In D3PLOT

LOCATE Identifying a curve's item visually in D3PLOT

The **LOCATE** function lets you right-click on a curve in the T/HIS window to identify the item associated with it. D3PLOT will then draw cross-hairs through that item in the graphics window. In this example node 1345 has been identified

.



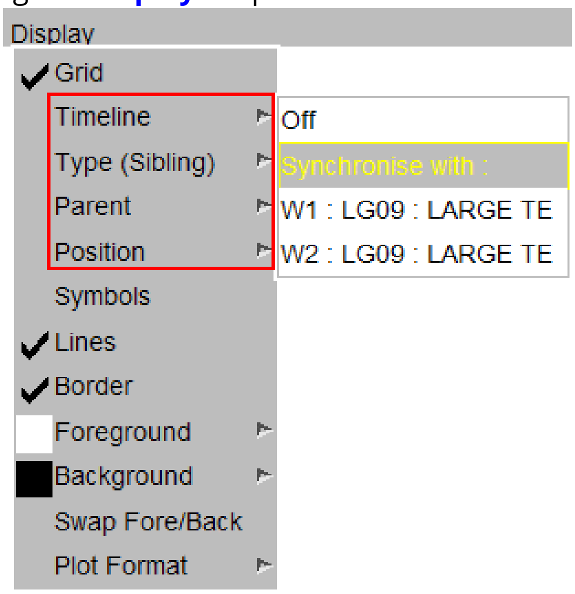
Only one item can be located at a time: each new pick supersedes the previous one. To turn off the crosshairs deselect **LOCATE** in the T/HIS panel.

9.13.3. Setting Properties for Individual Graphs

Setting Properties for Individual Graphs

The options in the T/HIS Global Command menu always apply to all of the T/HIS graphs that are currently active.

Properties for individual graphs can be set using the **Display** drop down menu at the top of each T/HIS graph.



Timeline	This option can be used to turn on and off the display of the Timeline and the D3PLOT window controlling the timeline for a single T/HIS graph.
Type	This option control the position of the T/HIS graph in relation to other windows (see Graph Type for more details)
Parent	Set the parent for CHILD windows (see Parent for more details)
Position	Set the position for CHILD windows (see Position for more details)

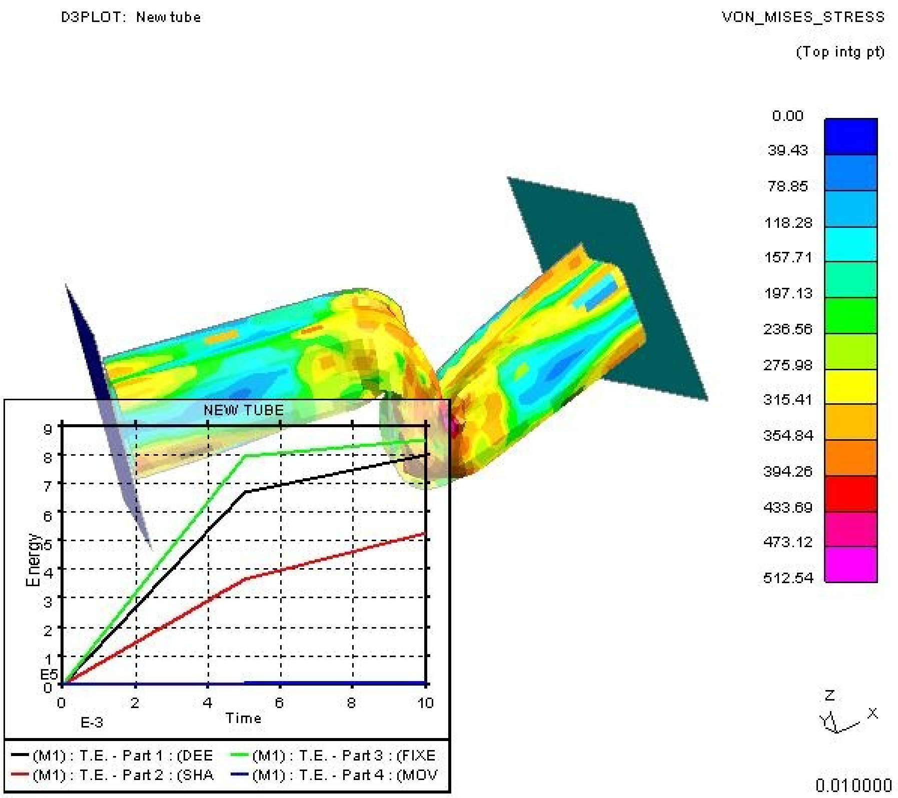
9.13.4. Using IMAGES to Capture Linked Images to File

Using IMAGES to capture linked images to file

The **IMAGES** command in D3PLOT is capable of capturing images containing both D3PLOT and T/HIS windows, (see IMAGES for more details.)

As T/HIS windows can be located within D3PLOT windows the transparency of T/HIS windows can be adjusted so that the underlying D3PLOT image can be seen through the T/HIS window.

The figure below shows an example of a 50% transparent "docked" T/HIS image overlying the D3PLOT one. In the undocked or "sibling" cases a composite image will be the size of the rectangular bounding box required to enclose the selected windows.



9.14. PRIMER: Sychronising with PRIMER

PRIMER: Sychronising with PRIMER

From D3PLOT 15.0 onwards, D3PLOT can be synchronised with PRIMER using a shared memory link. This means that a post-processing model that is open in D3PLOT can have its corresponding keyword file open in PRIMER, and information can be exchanged between the two codes. In addition graphics can be synchronised, so that the views of the model match in both codes.

By default no link takes place, but it can be opened in any of the following ways:

- A running D3PLOT session starts a new PRIMER session using the stipulated model.
- A running PRIMER session starts a new D3PLOT session using the stipulated model.
  and

- Once a link is established, in either of the modes above, further models can be opened and linked at will.

The link is symmetrical and bi-directional, with no concept of parent or child, and it can be closed at any time leaving both codes running autonomously. What you ***can't*** do at present is to link an autonomous, already running, D3PLOT or PRIMER session with another autonomous session.

9.14.1. The PRE Panel

The **PRE** panel

When running linked with PRIMER the Pre panel (invoked by pressing the **PRIMER** button) shows the current status of the link. In this example we have four models open in D3PLOT, and in this example:

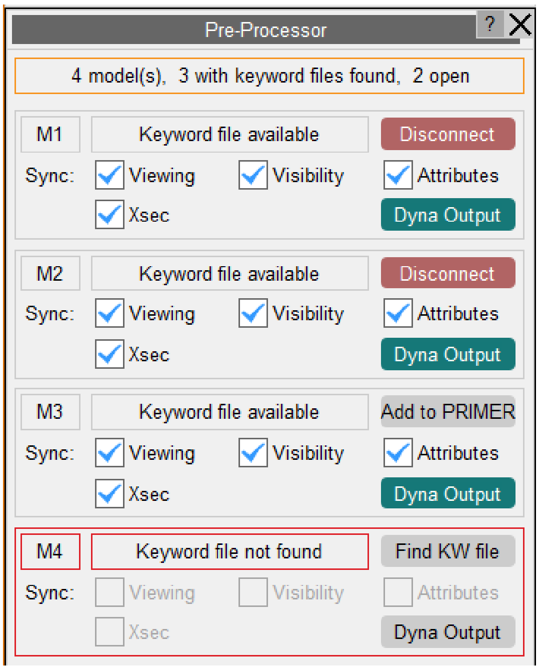
- Models 1 and 2 are currently open in PRIMER
- Model 3 is not open in PRIMER, but a keyword file has been found automatically.
- Model 4 is also not open in PRIMER, and D3PLOT has not found a keyword file automatically.

The file open/close options are

Option	Status of model	Action performed
Add to PRIMER	Not linked	D3PLOT has found a keyword file automatically, add this model to PRIMER
Find KW file	Not linked	D3PLOT cannot find a keyword file, browse for a filename manually
Disconnect	Linked	Model is linked with PRIMER session, disconnect it

There is a corresponding Post panel in PRIMER, with the same layout and functionality.

The button **Dyna Output** will open up an integrated PRIMER session and map the Dyna output tree panel. The error and warning messages from the dyna run will be shown in this panel. User can also **Edit** , **Sketch** entities and look at its **Xrefs** from this panel. These options are available only when the related keyword model file is located by D3PLOT. When you sketch parts and elements in PRIMER from this panel in PRIMER, these entities will also get sketched in D3PLOT graphics window as well. You can also change the timestep in D3PLOT corresponding to an error message using **Go to time** popup in the same Dyna Output tree panel.



Effects of linking and unlinking models in all cases:

- Linking or disconnecting a model does not affect that model's status in either programme, both D3PLOT and PRIMER will continue to run normally.
- Models may be disconnected and reconnected at will.
- When a model is deleted in D3PLOT it is implicitly disconnected in PRIMER, but will not be deleted from PRIMER. Similarly if a model is deleted in PRIMER is will be disconnected from D3PLOT, but not deleted.
- The link logic attempts to keep model numbers the same in both PRIMER and D3PLOT, however it is possible to defeat this by opening additional models in one programme but not the other. Doing so may cause the linkage to fail in some respects (so don't do it!)

The PRE panel can be opened or closed at will without affecting the status of linked models, it simply provides feedback about the current status and attributes of linked models.

9.14.2. Synchronising Attributes

Synchronising attributes.

It is possible to synchronise the following attributes across the link:

Attribute	What it does

Viewing	The current view: scale, orientation, position on screen, perspective settings. Includes the effect of dynamic viewing.
Visibility	Blanking and entity visibility settings
Attributes	Item colour, transparency and drawing mode (current, shaded, etc)
Xsec	Cut-sections: location, orientation, setting. Includes the effect of dragging the section.

Symmetry:

All the above attributes are symmetrical. For example if viewing is synchronised then a view change in D3PLOT will affect PRIMER, and one in PRIMER will effect D3PLOT.

Switching on/off

Each attribute type can be turned on/off independently for each model. The switches themselves are symmetrical: changing a setting in the Pre panel of D3PLOT will update the same setting in the Post panel in PRIMER.

Effects of multiple models

D3PLOT may put multiple models in different windows or the same window, but PRIMER places all models in the single window. This can lead to slightly strange behaviour since rotating only a single model (of several) in its own window in D3PLOT will affect all models in PRIMER, whereas rotating a model in PRIMER will affect all windows containing linked model in D3PLOT.

9.14.3. Synchronised Operations

Synchronised Operations

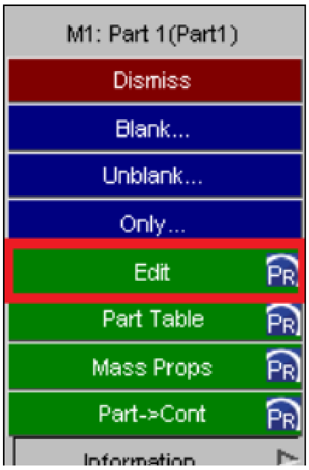
Both commands and data can be exchanged across the link using the following methods:

9.14.3.1. Quick Pick Edit

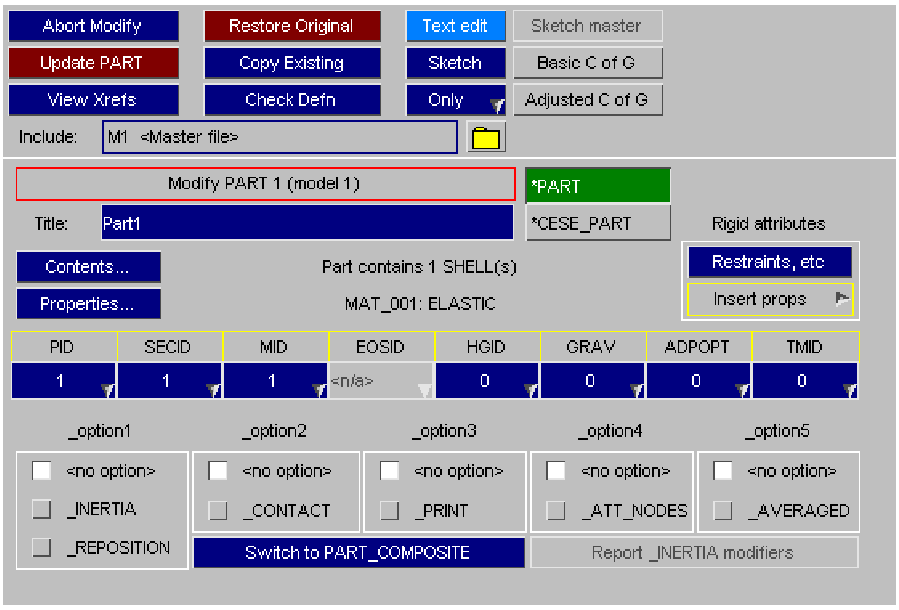
Quick Pick Edit

Like other Quick Pick commands this may be set as the current operation, or selected from a menu of choices (as shown here) in response to a right click.

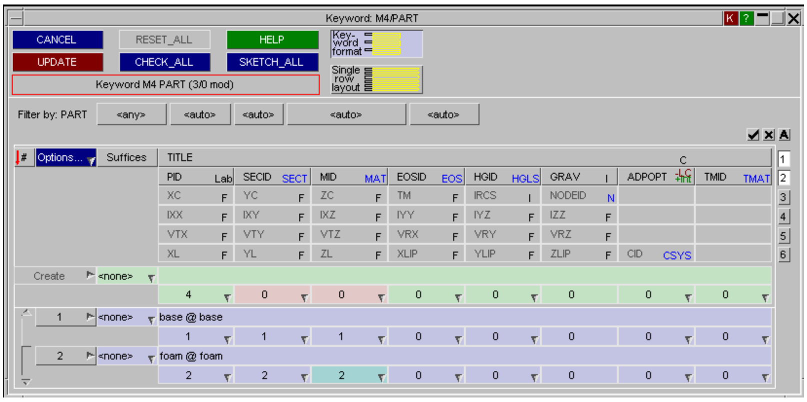
The editing window is launched in the linked PRIMER session, and the detailed action depends on how may items have been selected:



Picking a ***single*** item launches a singlescalar editor in PRIMER, for example here is the editor for a single part:



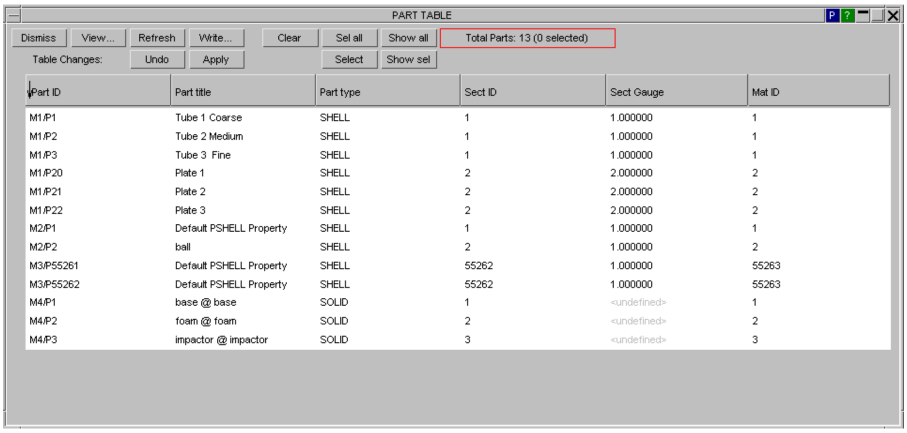
Picking ***multiple*** items launches a keyword editor, for example for multiple parts



9.14.3.2. Quick Pick Part Table

Quick Pick Part Table

All selected parts are launched in a Part Table in the linked PRIMER window, for example:

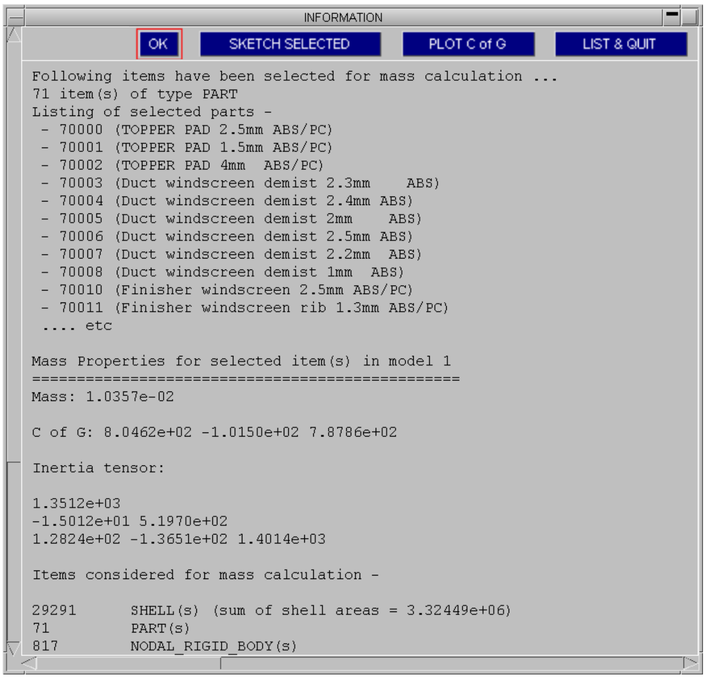


Part ID	Part title	Part type	Sect ID	Sect Gauge	Mat ID
M1/P1	Tube 1 Coarse	SHELL	1	1.000000	1
M1/P2	Tube 2 Medium	SHELL	1	1.000000	1
M1/P3	Tube 3 Fine	SHELL	1	1.000000	1
M1/P20	Plate 1	SHELL	2	2.000000	2
M1/P21	Plate 2	SHELL	2	2.000000	2
M1/P22	Plate 3	SHELL	2	2.000000	2
M2/P1	Default PSHELL Property	SHELL	1	1.000000	1
M2/P2	ball	SHELL	2	1.000000	2
M3/P55261	Default PSHELL Property	SHELL	55262	1.000000	55263
M3/P55262	Default PSHELL Property	SHELL	55262	1.000000	55263
M4/P1	base @ base	SOLID	1	<undefined>	1
M4/P2	foam @ foam	SOLID	2	<undefined>	2
M4/P3	impactor @ impactor	SOLID	3	<undefined>	3

9.14.3.3. Mass Properties Information

Mass Properties Information

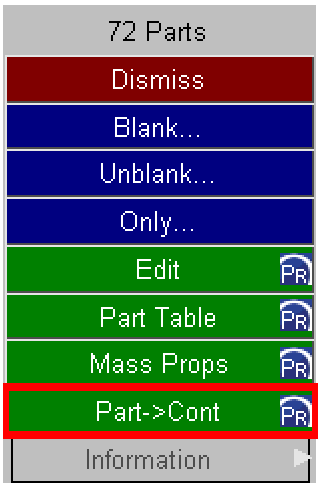
Mass information for all selected parts can be viewed in the mass information window in the linked PRIMER window, for example:

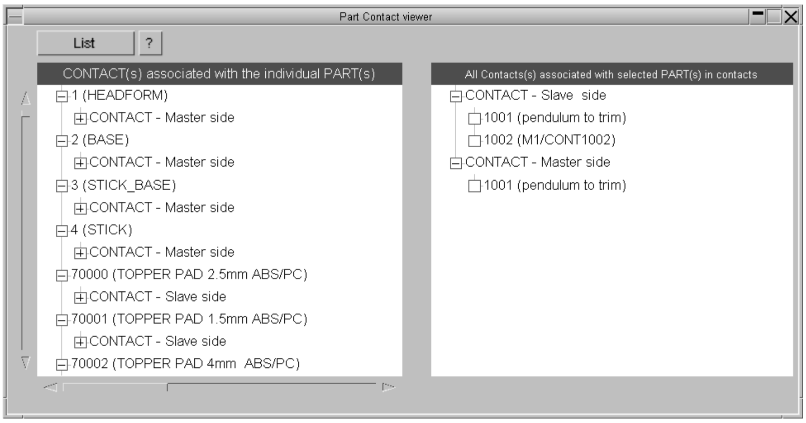


9.14.3.4. Part <-> Contact Information

Part<->Contact Information

Information regarding the contacts that selected parts are in can be viewed in the Part Contact viewer in the linked PRIMER window, for example:



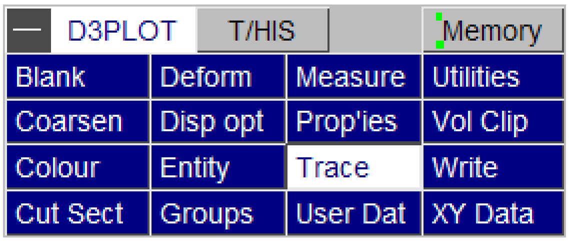


9.15. Trace

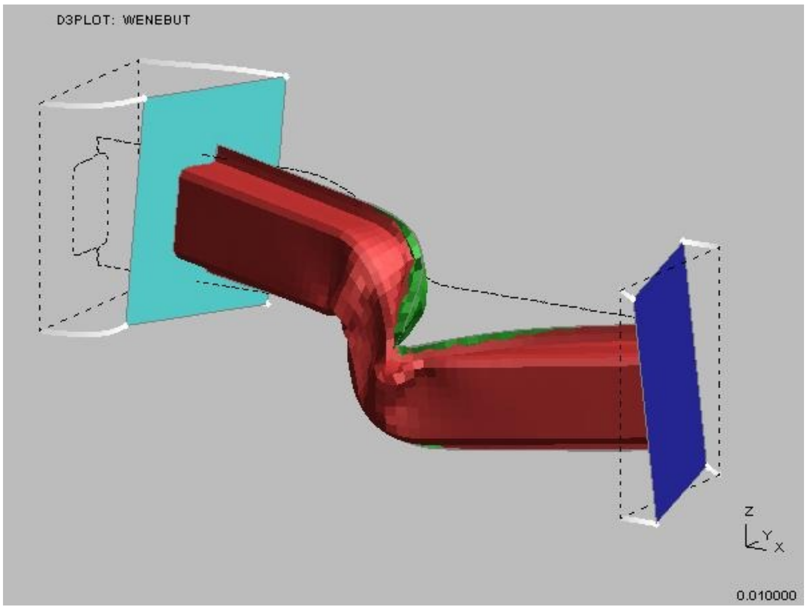
Trace

Trace adds lines ("traces") showing the motion history of selected Nodes, Airbag Particles and SPH elements.

The entities traced are user-selected, and the attributes of the trace lines are also selectable.



Here is an example of a crush tube on which the corner nodes of the plattens have been selected for traces. The outline undeformed geometry is also displayed to show how the trace paths display the path from original to current geometry.



9.15.1. The Trace Control Panel

The Trace control panel

The steps to be gone through are:

1. Select the entity type to trace from the **Trace type** popup (select from Node, Airbag Particle or SPH element).
2. Use **Create...** to define the entities for which you want traces. These can be typed in by label or screen picked.
3. Turn the tracer switch **ON** and traces will be added to your plots.
4. You can add more entities at any time using **Create...** , and likewise delete traces using **Delete...**

The meaning of the remaining controls is as follows. Click on an item to go to its detailed description.

Trace Type	Set the type of entity to select (Nodes, Airbag Particles and SPH elements)
Trace Colour	Sets the colour to be used. May be fixed or based on data values at the node.
Trace width	Trace line width. May be fixed or based on dat avalues

Trace length

How many points prior to the current node position are shown.

Solid / Faded line

The line can be solid along its length, or can fade out.

Line and symbol display

The line itself is always shown, but you can also choose to show the current and previous symbols

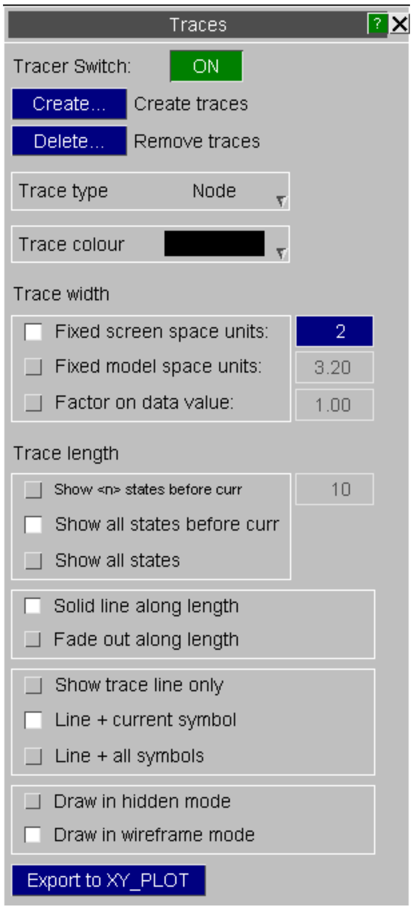
Hidden / wireframe mode

In hidden mode the lines are subject to hidden surface removal, ie they co-exist with other items.

In wireframe mode they are effectively drawn on top of any structure and will always be visible.

Export to XY_PLOT

Make a graph of the global co-ordinates of the nodes at each time-state.

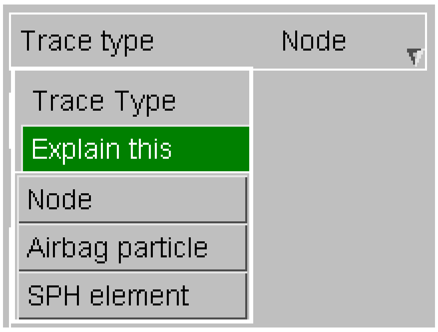


9.15.2. Trace Type

Trace Type

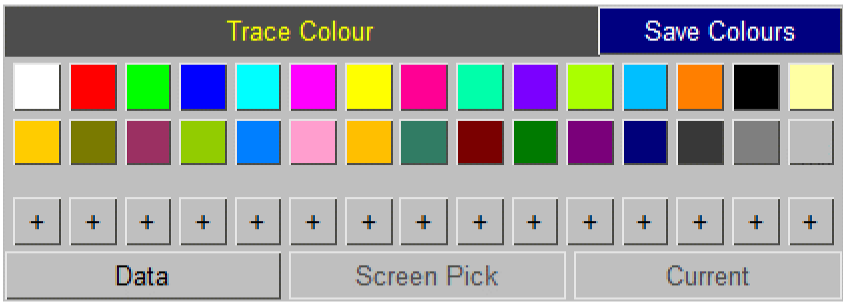
Sets the type of entity to select to trace (Nodes, Airbag Particles and SPH elements).

NOTE : Traces of nodes, airbag particles and SPH elements can be plotted at the same time, but selection of each type can only be done indivdually.



9.15.3. Trace Colour

Trace Colour

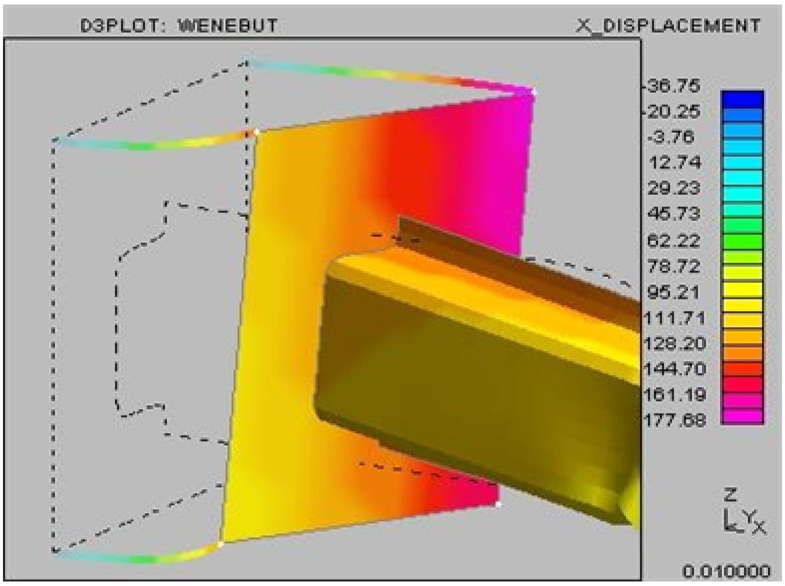


Traces can have a constant colour - either a standard core colours or a user-defined colour.

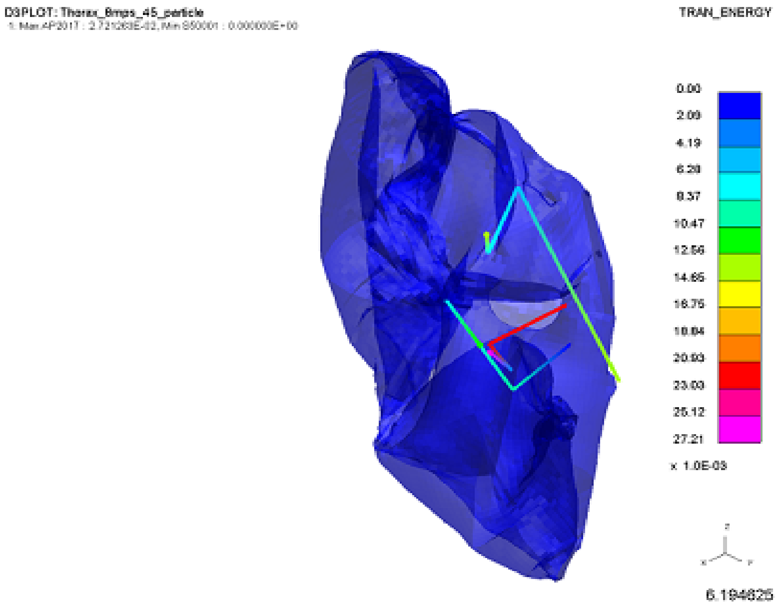
However Data requires some explanation:

If you have previously performed some sort of data-bearing plot that has generated values at nodes then **DATA** colours will draw traces bearing those colours at the selected entities. This allows you to show how values have changed over time.

For example the image below uses **DATA** mode to display colours showing X displacement on the example above. Here the traces have been added to a shaded image contour plot, but they would still be coloured if added to a non-data-bearing plot.



The image below shows another example using **DATA** mode to display Translational Energy for airbag particles.

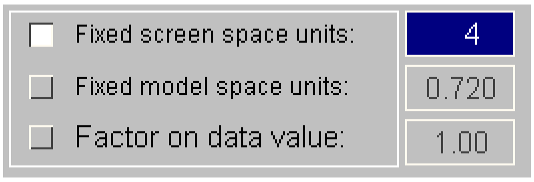


9.15.4. Trace Width

Trace Width

Sets the width of trace lines.

NOTE : Trace widths are clamped to lie between 1 and 25 screen units to prevent plots getting ridiculous in either of the last two cases



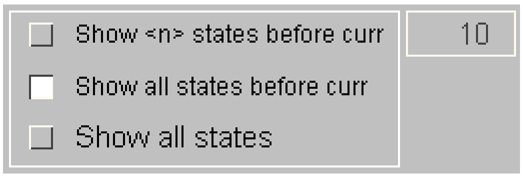
Fixed screen space units	Will give a fixed number of pixels width that will not change as the scale of the plot changes.
Fixed model space units	Will also give a fixed width, but in the actual units of the model, so this ***will*** change as the image scale changes.
Factor on data value	Will multiply the data value at the entity to produce a quantity that is treated as a screen space unit, so it will not change with image scale

9.15.5. Trace Length

Trace Length

Specifies how many points before the current one are shown in a trace line.

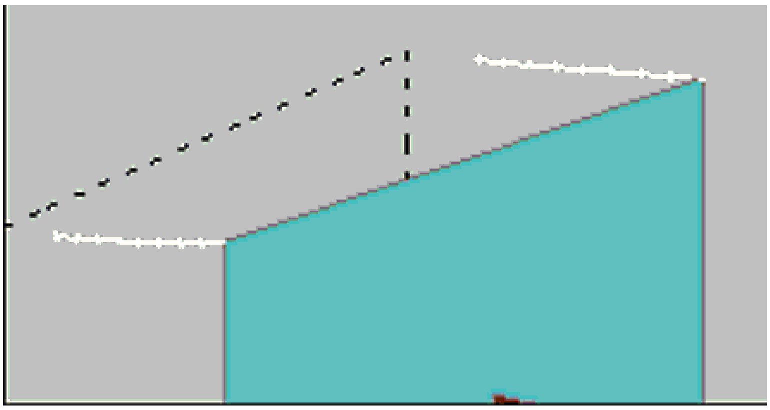
This is best demonstrated by example. In all cases point symbols have been turned on to make clear what is being displayed.



Show <n> states before curr

(Here <n> = 8)

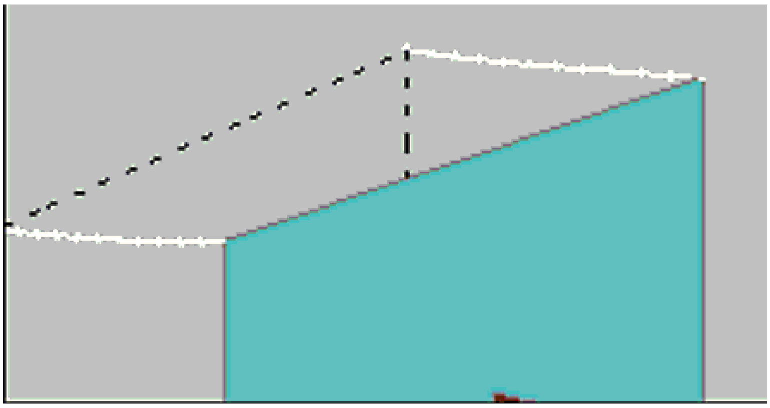
This shows the 8 states prior to the previous one.



If fewer than <n> states are available then only those between the current and state #0 will be shown.

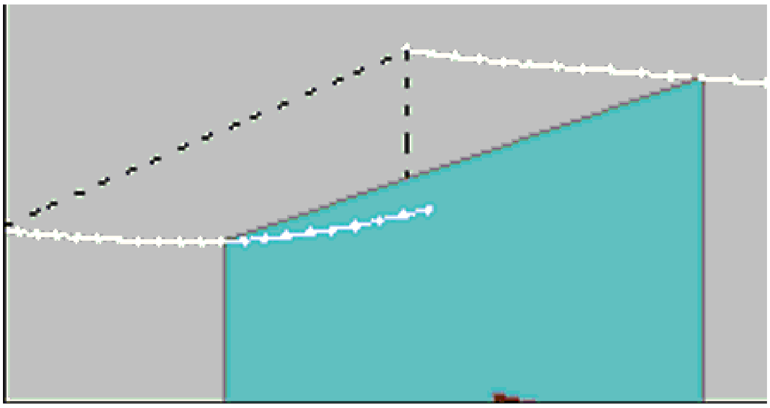
Show all states before curr

This shows all states from #0 to the current one.



Show all states

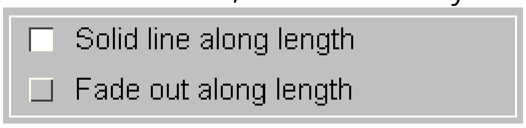
This shows all states, including those both before and after the current one.



9.15.6. Solid / Faded OutLine

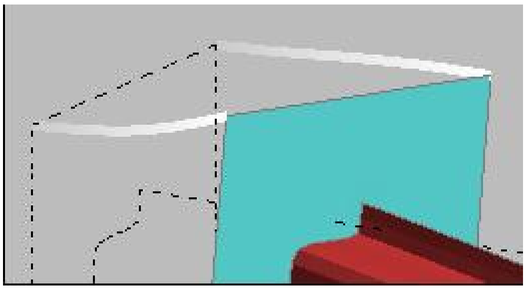
Solid / Faded out line

Determines whether lines are drawn solid, or whether they fade out as they get "older"



Solid lines are obvious (all three plots above use them).

Here is an example of a plot fading out along its length, showing that it gives a good visual indication of the "age" of each point compared to the current one.

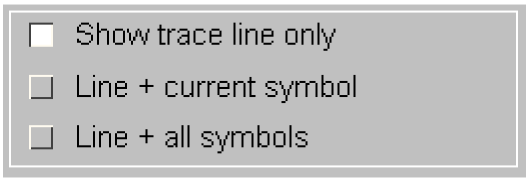


9.15.7. Symbol Display

Symbol Display

Determines whether symbols are shown, and at which positions.

"Symbols" are a diamond shape on the line at a data point.



Show trace line only:

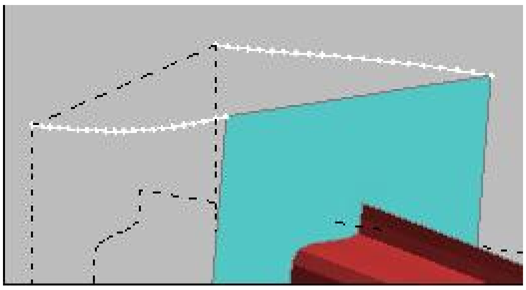
No symbols are drawn, and only the line is shown.

Line + current symbol:

The line is drawn, and a symbol is shown at the current point only.

Line + all symbols

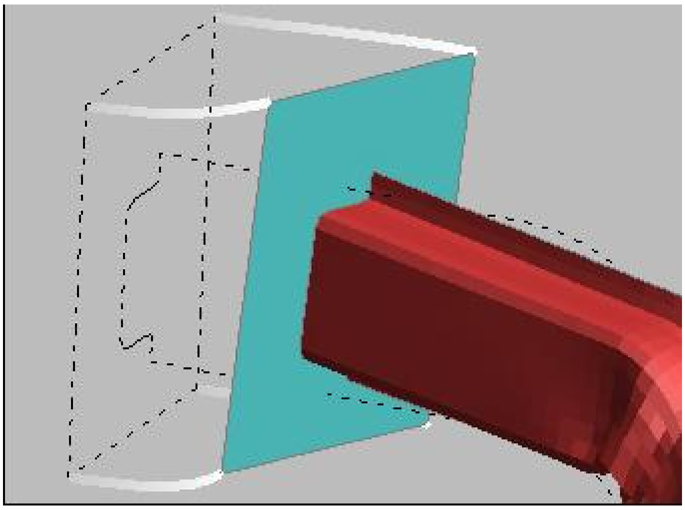
Symbols are drawn at all points, as shown here.



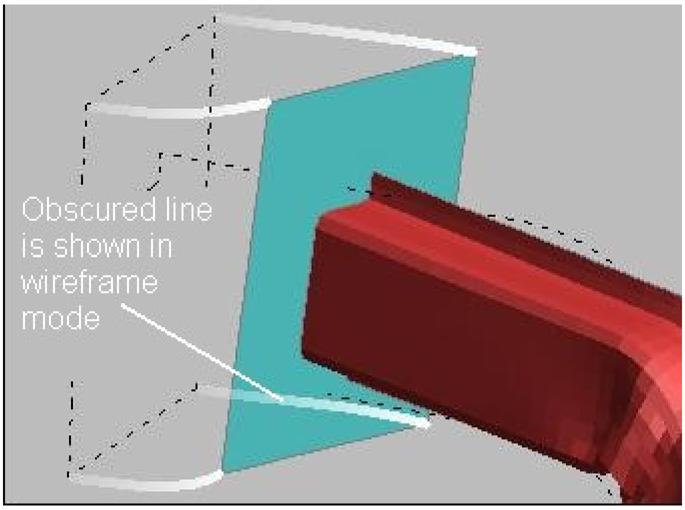
9.15.8. Hidden vs Wireframe Mode

Hidden vs Wireframe mode

In "Hidden" mode trace lines occupy the same 3D space as the model, and may become obscured by structure.



In "Wireframe" mode trace lines are effectively drawn above the model and will always be visible.



9.15.9. XY Plot

XY Plot

Make a graph of the global co-ordinates of the traces at each time-state. The plots can be saved as curve files or transferred to T/HIS if the T/HIS link is open.

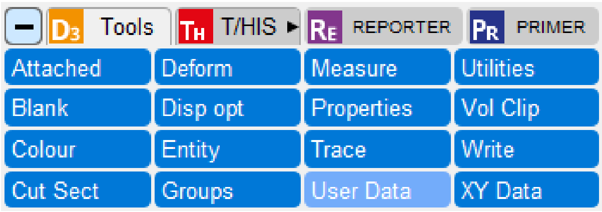
9.16. User Data

User Data

"User Data" creates and manages user-defined data components.

These allow completely arbitrary data for nodes and elements to be processed. The data may be generated externally, or computed locally from existing data already present in the database.

Special "binary" user-defined data components may be created and manipulated via the JavaScript API.



9.16.1. Description of User Defined Data Components

Description of User-defined Data components

Conventional data components in D3PLOT are typically nodal displacements, velocities and accelerations, and element stresses, but are limited to what is present in the database that has been read.

User-defined components take this a stage further and allow you to define any number of new data components of the following nine types:

Entity type	**Class of data**
Nodes	**Scalar** : A single value at nodes
	Vector : A [x,y,z] data vector at nodes
Solids, shells, thick shells	**Scalar** : A single value at elements
	Tensor : A [xx,yy,zz,xy,yz,zx] data tensor at elements
Beams	**Scalar** : A single value at beams
	Vector : A [x,y,z] data vector at beams
Other	**Scalar** : A single value at elements
	Vector : A [x,y,z] data vector at elements
	Tensor : A [xx,yy,zz,xy,yz,zx] data tensor at elements

Each user-defined component must fall into one of the nine classes in the second column above. It is not possible to mix classes in a single component.

Once a component has been defined it may be used exactly like "normal" data in any valid context within D3PLOT: data plotting, written value extraction, XY plotting, etc. Where vector and tensor components are defined, standard operations such as extracting vector direction and magnitude, or deriving sub-components such as principal or von Mises sub-components, may be performed.

A user-defined component may have its name, content or calculation method changed at any time; however it may not be swapped between classes. (Although it may be deleted and re-created as a new class.)

User-defined components are treated as "programme-wide" attributes, meaning that the same components apply to all models read into the database. If you need to apply separate components to different models it will be necessary to create new user-defined components with different attributes for the various models.

User-defined components must have names that are unique within D3PLOT. This means that not only must their names not clash with each other, but they must also not be the same as "standard" components: for example you will not be permitted to create a user-defined component called "strain".

There are four types of user-defined component in D3PLOT. The first three are created interactively or externally, and the last is used only when using the JavaScript API. In summary they are:

<table>
<tr><td>(1) Read from file</td><td>Data is read from externally supplied ASCII data file(s)</td><td rowspan="3">These three methods are easy to use, and in particular Simple Formulae may be created interactively.

They are suitable for simple processing on a per node or element basis.</td></tr>
<tr><td>(2) Simple formula</td><td>Data is calculated internally from a "simple" formula using standard internal data and arithmetic.</td></tr>
<tr><td>(3) JavaScript file</td><td>Data is calculated using a small JavaScript file.</td></tr>
<tr><td>(4) User-defined Binary (UBIN)</td><td colspan="2">Data is created via the JavaScript API (see The JavaScript Interface), not to be confused with method (3) above

This is a much more powerful method than the three above, in that data can be extracted in an arbitrary fashion and written likewise, however it is not interactive. It is described briefly in "User Defined Binary" (UBIN) Components Generated from the JavaScript API below,</td></tr>
</table>

	and the API functions are described fully in the JavaScript API Reference Manual

9.16.2. Creating a New User-Defined Component

Creating a new user-defined component

The steps to go through are:

1. Use **New...** to start a new component definition

2. Choose one of the possible data classes:

Node scalar	A single scalar value at nodes
Node vector	A [x,y,z] vector at nodes (3 values)
So/Sh scalar	A single scalar value at solids, shells, thick shells
So/Sh tensor	A [xx,yy,zz,xy,yz,zx] tensor at solids, shells, thick shells (6 values)
Beam scalar	A single scalar value at beams
Beam vector	A [x,y,z] vector at beams (3 values)
Othr scalar	A single scalar value at elements
Othr vector	A [x,y,z] vector at elements (3 values)
Othr tensor	A [xx,yy,zz,xy,yz,zx] tensor at elements (6 values)

3. Define a component name. A default name will be generated, but any name (no more than 30 characters long) may be used, **so long as it is unique** : it must not conflict with any built-in component names, or any other user-defined component names.

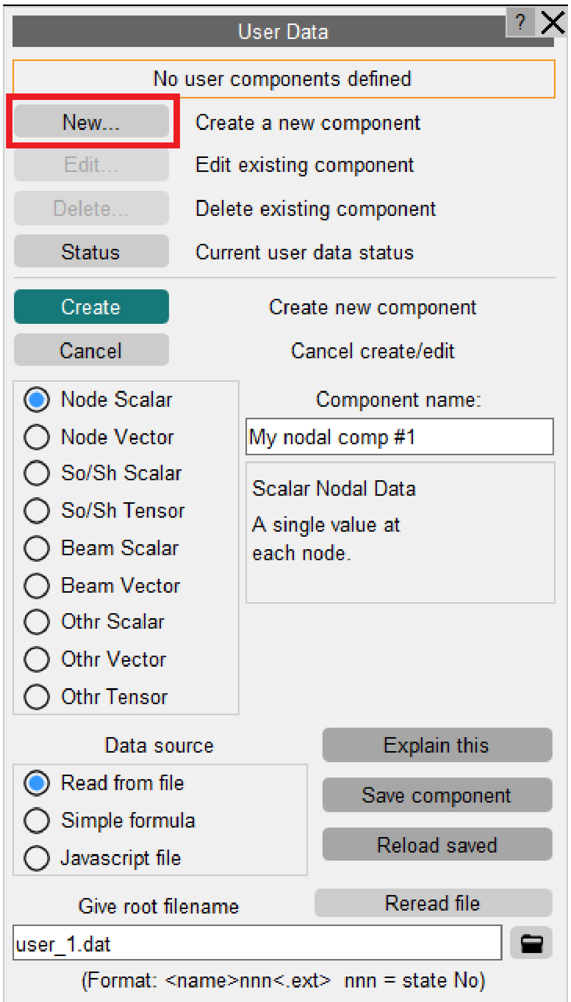
4. Define a data source from one of:

Read from file	Data is read from externally supplied ASCII data file(s)
Simple formula	Data is calculated internally from a "simple" formula using standard internal data and arithmetic.
JavaScript file	Data is calculated externally using a JavaScript file.

5. Define a "root" data filename, simple formula or JavaScript filename depending on the data source.

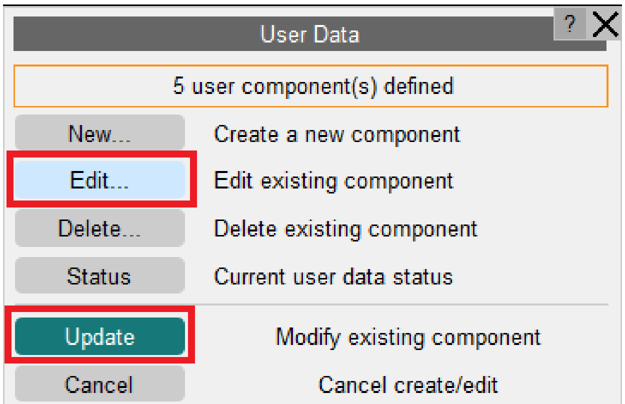
6. Finally **CREATE** the new component to save it.

There is also a category of "User Defined Binary Components", referred to as UBIN, that may only be generated by the separate JavaScript interface to D3PLOT (not to be confused with the "JavaScript file" data source referred to here). See the JavaScript API documentation for more information about these.



9.16.3. Editing an Existing User-Defined Component

Editing an existing user-defined component



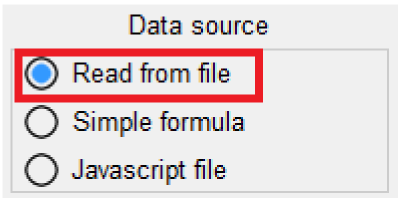
An existing component may be edited at any time.

- Click on **Edit...**
- Choose the component to be edited from the menu
- Change its attributes.
- Click on **Update** to save the changes

The component name and data source may be changed ad lib, but the basic class of the component (Node/SoSh/Beam/Other and Scalar/Vector/Tensor) cannot be altered. To change the component class it is necessary to delete it and then recreate it using the new class.

9.16.4. "Read From File"

"Read from File"



This method assumes that data in the appropriate format has been generated externally and saved in ASCII format data files.

Filename syntax

A separate file is required for each state, and its name is determined from the "root" filename supplied. This must be of the form

<name>* nnn *<.ext> Where:

<name>	Is an arbitrary prefix	For example a valid filename might be"nodal_1.dat" (decomposes to < **nodal_** >, < **1** >, < **.dat** >)

nnn	Is one or more digits.	The actual number used in the "root" filename is not material, it must just be a digit from 0 - 9.
<.ext>	Is an arbitrary extension	

For each state the digit <nnn> is replaced with the number of the state, so in the example above D3PLOT would look for the file sequence:

`nodal_1.dat` for state #1, **`nodal_2.dat`** for state#2, and so on.

The number **`nnn`** can have up to 10 leading zeros, so both **`nodal_1.dat`** and **`nodal_0001.dat`** would be acceptable filenames, and the number of leading zeros does not have to be the same for each state.

If a file for a given state is not found then values of 0.0 will be used for that state.

File content and structure

In all files:

- Blank lines are ignored.
- Lines starting with #, % or $ are treated as comment lines and are ignored.
- Input is not case-sensitive: any mixture of upper and lower case can be used.
- Numeric input is "free": no particular field width or column number is implied.
- Field separators may be spaces, tabs or commas.
- Each row of data must be on a single line.
- Maximum line width is 256 characters.

The formats of the various data types are as follows:

Scalar data		
One of	**`NODE SCALAR`** **`SOLID SCALAR`** **`BEAM SCALAR`** **`SHELL SCALAR`** **`TSHELL SCALAR`**	A new header can appear at any time to start a new data category.
Optionally:	**`DEFAULT`** ***<value>***	A default of 0.0 is assumed if this line is not present
Optionally:	**`SURFACE TOP`** or **`MIDDLE`** or **`BOTTOM`** or **`LAYER`** ***<layer no>*** For models with fully integrated elements and data written at each on plan integration point (MAXINT -ve), the on plan point to read data into can be specified using the syntax:	Ignored for Node, Solid or Beam data. For shells the current surface is assumed unless a SURFACE or LAYER line is defined.

	SURFACE TOP ***@<on plan>*** or **MIDDLE@** ***<on plan>*** or **BOTTOM** ***@<on plan>*** or **LAYER** ***@<on plan> <layer no>***	The first on plan point is used if it is not defined.
Followed by any number of lines	***<label> <value>*** ***<label> <value>***	If <label> is not found in the model the line is ignored. Entities which don't have a value line are given the default value.
Coordinate system	*none is implied*	Does not apply to scalar values.

Example of a nodal scalar data file:

```
NODE SCALAR

default 400.0

12 1.382457E+02
13 -4.655358E+01
14 -2.706973E+02
15 -2.615501E+02
16 -1.364710E+02
17 8.553621E+00
```

Example of a shell scalar data file:

```
SHELL SCALAR

default 10.0

surface top

1 1.23456e+02
2 2.34578e-02
```

Vector data

One of	**NODE VECTOR** **BEAM VECTOR**	A new header can appear at any time to start a new data category. Vector data is not valid for solids, shells or thick shells
Optionally:	**DEFAULT** ***<3 values: x,y,z>***	A default of (0.0, 0.0, 0.0) is assumed if this line is not present
Followed by any number of lines	***<label> <3 values: x,y,z>*** ***<label> <3 values: x,y,z>***	If <label> is not in the model found the line is ignored. Entities which don't have a value line are given the default vector value.

Coordinate system:	**Nodal data** is assumed to be in the **global cartesian system** **Beam data** is assumed to be in the **element local system**	These two conventions match the raw data in LS-DYNA database files.

Example of a nodal vector data file:
 	
`NODE VECTOR`
 	
`default 1.0 2.0 3.0`
 	
`1 6.988275E+02 1.032284E+02 -1.468281E+02`
`2 7.038895E+02 7.875581E+01 -1.475109E+02`
`3 7.089514E+02 5.428316E+01 -1.481938E+02`
`4 7.140133E+02 2.981051E+01 -1.488766E+02`
`5 7.190753E+02 5.337874E+00 -1.495595E+02`
`6 7.241373E+02 -1.913478E+01 -1.502424E+02`
`7 7.291992E+02 -4.360743E+01 -1.509252E+02`
`8 7.342611E+02 -6.808007E+01 -1.516081E+02`
`9 7.393231E+02 -9.255272E+01 -1.522909E+02`
`10 7.375262E+02 -9.361953E+01 -1.273784E+02`		

Tensor data					
One of	**SOLID** **SHELL** **TSHELL**	Followed by one of	**TENSOR** **TENSOR_UPPER** **TENSOR_LOWER**	TENSOR implies linear data order TENSOR_UPPER implies upper triangular order TENSOR_LOWER implies lower triangular order	[xx, yy, zz, xy, yz, zx] [xx, xy, xz, yx, yy,

				(The tensor is implicitly symmetric)	zz] [xx, yx, yy, zx, zy, zz]
Optionally:	**DEFAULT** ***<6 values>***			A default of [0.0, 0.0, 0.0, 0.0, 0.0, 0.0] is assumed if this line is not present If a default is supplied it must be in the order implied above.	
Optionally:	**SURFACE TOP** or **MIDDLE** or **BOTTOM** or **LAYER** ***<layer no>*** For models with fully integrated elements and data written at each on plan integration point (MAXINT -ve), the on plan point to read data into can be specified using the syntax: **SURFACE TOP** ***@<on plan>*** or **MIDDLE@** ***<on plan>*** or **BOTTOM** ***@<on plan>*** or **LAYER** ***@<on plan> <layer no>***			Ignored for Solid data. For shells the current surface is assumed unless a SURFACE or LAYER line is defined. The first on plan point is used if it is not defined.	
Optionally one of	**GLOBAL** **LOCAL_DYNA** **LOCAL_NASTRAN_2D** **LOCAL_NASTRAN_3D**			Default, and need not be specified, tensor data is normally assumed to be global Tensor, all six 3D terms, is in LS-DYNA local element axis system Tensor, only three 2D terms, is in NASTRAN local element axis system Tensor, all size 3D terms, is	

		in NASTRAN local element axis system
	This option permits data to be defined externally in the element local coordinate system. It will be converted to global when read, and stored internally in the global system. ***The definitions of element local coordinate systems in DYNA and NASTRAN are not the same*** , so make sure that you specify the correct one for your data source in order to obtain the correct transformation back to the global system. A 2D tensor, the **LOCAL_NASTRAN_2D** case, reads only 3 columns of [xx, yy, xy] data, in that order, and is suitable for "plane stress" thin shell data. The "missing" 3D terms (yz, zx, zz) will be set to zero internally to create a full 3D local tensor, then the result will be converted to the global cartesian system as for the other local cases above.	
Followed by any number of lines	***<label> <6 values>*** ***<label> <6 values>***	If <label> is not found in the model the line is ignored. Data must be in the tensor order implied by the header above Entities which don't have a value line are given the default values.
Coordinate system	The **global cartesian system** is assumed, but conversion from external local to internal global is possible via a **LOCAL_ *xxx*** option.	This matches the raw data in the LS-DYNA database

Example of a shell tensor data file:

```
SHELL TENSOR

Default 1.0 2.0 3.0 4.0 5.0 6.0
Layer 1

1 1.1567E+02 -3.9286E+01 -1.2514E-01 -1.2733E+01 3.8389E+00 -
4.6773E+00
2 7.0594E+00 3.7640E+00 -1.1621E+00 5.0975E+01 9.0807E+00 -9.8825E-
01
```

```
3 1.0105E+02 1.4159E+01 9.5627E-01 1.1344E+01 -3.2170E+00 -
8.7537E+00
4 1.4473E+00 -5.2854E+01 1.7974E+00 2.5017E+01 -1.0200E+01 -
5.3932E+00
5 8.9096E+01 -4.0171E+01 3.4451E-01 -4.2868E+00 3.0106E+00 -
6.8030E+00
6 7.1164E+00 9.4949E+00 -1.1046E+00 4.2835E+01 4.2636E+00
1.6420E+00
7 2.4371E+02 2.0140E+01 -1.7336E+01 -7.3324E+01 -3.4727E+00 -
1.1746E+01
8 2.1562E+02 3.8205E+01 2.8096E+00 -4.1887E+01 1.1632E+01 -
1.0129E+01
9 2.2813E+02 2.3411E+01 1.8917E+01 -6.7636E+01 1.1483E+01 -
1.0553E+01
10 1.9249E+02 -2.1300E+01 -1.3462E+00 -4.9531E+01 -5.8716E+00 -
8.7624E+00
```

Example of a shell tensor data file that is ex-Nastran data defined in the element local system.
Note that only XX, YY and XY terms are defined in this _2D case.

```
SHELL TENSOR
LOCAL_NASTRAN_2D

Surface top

1 6.988275E+02 1.032284E+02 -1.468281E+02
2 7.038895E+02 7.875581E+01 -1.475109E+02
3 7.089514E+02 5.428316E+01 -1.481938E+02
4 7.140133E+02 2.981051E+01 -1.488766E+02
5 7.190753E+02 5.337874E+00 -1.495595E+02
6 7.241373E+02 -1.913478E+01 -1.502424E+02
7 7.291992E+02 -4.360743E+01 -1.509252E+02
8 7.342611E+02 -6.808007E+01 -1.516081E+02
9 7.393231E+02 -9.255272E+01 -1.522909E+02
10 7.375262E+02 -9.361953E+01 -1.273784E+02
```

Multiple headers are permitted in a file

A file can contain any number of data headers of the relevant type, for example a file for "Solid and Shell scalar data" can contain multiple headers for both solid and shell element types, for example the following file contains scalar data for both solid and shell elements:

```
SHELL SCALAR
default 10.0
surface top
```

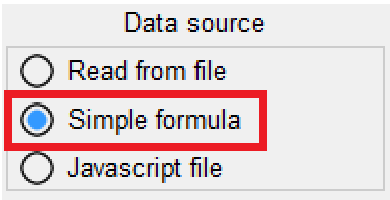
```
1 1.23456e+02
2 2.34578e-02

SOLID SCALAR
default 20.0
surface top

1 3.45678e+02
2 4.56789e-02
```

File reading of a particular type of data will continue until either a new header is found or [end of file] is encountered.

If a header that does not match the type of data expected is encountered (for example the data component is NODE SCALAR, but a BEAM VECTOR header is encountered) then a warning message will be written and the mis-matched data will be skipped. Whilst this is not a fatal error, it is recommended that you do not mix different data types in the same file. This can cause confusion for other users, and could cause D3PLOT to perform slowly due to reading large amounts of unecessary data.

9.16.5. "Simple Formula"

"Simple Formula"



This method calculates results from a "simple" mathematical formula using built-in data components and standard maths operations.

All calculations are performed using floating point arithmetic and produce a single result, thus:

- Scalar components are evaluated from a single formula
- Vector components use three (independent) formulae: one for each of [x,y,z]
- Tensor components use six formulae, one for each of[xx,yy,zz,xy,yz,zx]

Syntax rules for Simple Formulae

Formulae follow typical "computer programming" syntax, using commonly understood operators.

Operators supported:

*, /, +, -	Multiply, divide, plus and minus	* and / take precedence over + and -
** and ^	To the power of. Thus "x**2" is "x squared"	Note that "^" is Fortran syntax, ***not*** the C "exclusive OR" operator.
%	Modulus operator (x % y is remainder of x / y)	There is also a "mod" maths function.
(...)	Brackets may be nested to any (reasonable) level	(...) take precedence over all other operators
Note: all arithmetic is floating point, thus both "5.0 / 2.0" and "5 / 2" will both deliver the answer 2.5		

A formula must be a single line, up to 256 characters long.

Standard D3PLOT "built in" data components supported

Category	Acronym	Description	Acronym	Description
Nodal data	**bx** **by** **bz** **cx** **cy** **cz** **dx** **dy** **dz** **dm** **vx** **vy** **vz** **vm** **ax** **ay** **az** **am**	Basic X coordinate Basic Y coordinate Basic Z coordinate Current X coordinate Current Y coordinate Current Z coordinate X displacement Y displacement Z displacement Displacement magnitude X velocity Y velocity Z velocity Velocity magnitude X acceleration Y acceleration Z acceleration Acceleration magnitude	**rdx** **rdy** **rdz** **rdm** **rvx** **rvy** **rvz** **rvm** **rax** **ray** **raz** **ram** **temp** **tbot** **tmid** **ttop** **tfx** **tfy** **tfz** **tfm** **dtdt**	X rotation displacement Y rotation displacement Z rotation displacement Rotation displacement magnitude X rotation velocity Y rotation velocity Z rotation velocity Rotation velocity magnitude X rotation acceleration Y rotation acceleration Z rotation acceleration Rotation acceleration magnitude Temperature Nodal (shell)

				bottom surface temperature Nodal (shell) middle surface temperature Nodal (shell) top surface temperature X temperature flux Y temperature flux Z temperature flux Temperature magnitude dTemp / dTime
Solid and shell data	Stress and strain tensor-derived data is extracted in the global frame of reference unless the " `l` " (for "local") suffix is added to the component acronym, in which case it is extracted in the element local system. See below for how the Frame of Reference transformation applies to Simple Formula components.			
Global tensor components	**sxx** **syy** **szz** **sxy** *or* **syx** **syz** *or* **szy** **szx** *or* **sxz**	XX stress YY stress ZZ stress XY stress (*symmetric*) YZ stress (*ditto*) ZX stress (*ditto)*	**exx** **eyy** **ezz** **exy** *or* **eyx** **eyz** *or* **ezy** **ezx** *or* **exz**	XX strain YY strain ZZ strain XY strain (*symmetric*) YZ strain (*ditto*) ZX strain (*ditto*)
Element Local tensor components	**sxxl** **syyl** **szzl** **sxyl** *or* **syxl** **syzl** *or* **szyl** **szxl** *or* **sxzl**	XX stress YY stress ZZ stress XY stress (*symmetric*) YZ stress (*ditto*) ZX stress (*ditto)*	**exxl** **eyyl** **ezzl** **exyl** *or* **eyxl** **eyzl** *or* **ezyl** **ezxl** *or* **exzl**	XX strain YY strain ZZ strain XY strain (*symmetric*) YZ strain (*ditto*) ZX strain (*ditto*)

Non-directional components derived from tensor data	**svon** **smax** **smid** **smin** **sav** **sms** **evon** **emax** **emid** **emin** **eav** **ems**	von Mises stress Max principal stress Mid principal stress Min principal stress Average stress (pressure) Max shear stress von Mises strain Max principal strain Mid principal strain Min principal strain Average strain (pressure) Max shear strain	**epl** **erate** **pemag** **engmaj** **engmin** **engthk**	Effective plastic strain Strain rate Plastic strain magnitude Engineering Major strain Engineering Minor strain Engineering Thickness strain
Non-directional components from Nastran OP2 file	**sen** **senp** **send** **ken** **kenp** **kend**	Strain energy Strain energy percentage Strain energy density Kinetic energy Kinetic energy percentage Kinetic energy density	**enl** **enlp** **enld**	Energy loss Energy loss percentage Energy loss density
Shell only data	rfx rfy rfxy rqx rqy rmx rmy rmxy s2max s2min e2max e2min eratio	Fx resultant force Fy resultant force Fxy resultant force Qx resultant force Qy resultant force Mx resultant moment My resultant moment Mxy resultant moment 2D (in plane) max princ stress 2D (in plane) min princ stress 2D (in plane) max princ strain 2D (in plane) min princ strain	**thk** **area** **eden** **hgen** **tstp** **mass** **madd**	Shell Thickness Shell Area Internal energy density Hourglass energy Timestep Mass Added mass

		2D (in plane) princ strain ratio		
Solid only data	erate	Strain rate	vol rvol	Element Volume Element Relative volume
Beam only data	**bfx** **bfy** **bfz** **bfr** **bmxx** **bmyy** **bmzz** **brm** **bsxx** **bsyx** *or* **bsxy** **bszx** *or* **bszx** **bep** **beax** **bsax** **bpe1** **bpe2** **bry1** **bry2** **brz1** **brz2** **bmy1** **bmy2** **bmz1** **bmz2** **baen** **bie** **brxx**	Axial force YY shear force ZZ shear force Resultant force XX torsional moment YY bending moment ZZ bending moment Resultant moment XX axial stress YX shear stress (*symmetric*) ZX shear stress (*ditto*) Plastic strain Axial strain Total axial strain Plastic energy at end 1 Plastic energy at end 2 Y rotation end 1 Y rotation end 2 Z rotation end 1 Z rotation end 2 Y moment end 1 Y moment end 2 Z moment end 1 Z moment end 2 Axial energy Internal energy Torsional rotation	**bbed** **baed** **bied** **bsen** **bsenp** **bsend** **bken** **bkenp** **bkend** **benl** **benlp** **benld**	Bending energy density Axial energy density Internal energy density Strain energy Strain energy percentage Strain energy density Kinetic energy Kinetic energy percentage Kinetic energy density Energy loss Energy loss percentage Energy loss density
"Extra" and ALE data	**sox** *n* **shx** *n*	Solid Extra component #n Shell Extra component #n	**ammg** *n* **amss** *n* **adens** **adomf**	Ale Multi-Matl group #n Mass of MM group #n

				Ale density Ale dominant fraction
Contact data	csn cst csx csy carea	Contact Normal Stress Contact Tangential stress Contact local X stress Contact local Y stress Contact segment area These are contact segment data averaged at nodes	cfgx cfgy cfgz cflx cfly cflz cfm	Contact global X force Contact global Y force Contact global Z force Contact local X force Contact local Y force Contact local Z force Contact force magnitude
LSDA (binout) data (Only available if both a ZTF file and a binout file have been read.)	**sw_f** **sw_s** **sw_trsn** **sw_fail** **sw_time** **sp_f** **sp_e** **sp_m** **sp_r** **xsec_f_x** **xsec_f_y** **xsec_f_z** **xsec_m_x** **xsec_m_y** **xsec_m_z** **xsec_a**	Spotweld axial force Spotweld shear force Spotweld torsion moment Spotweld failure Spotweld failure time Spring axial force Spring elongation Spring torsional moment Spring rotation Database X-sect X force ditto Y force ditto Z force ditto X moment ditto Y moment ditto Z moment Database X-sect area	**sb_f** **sb_l** **sr_p** **rt_f** **rt_p** **spc_f_x** **spc_f_y** **spc_f_z** **spc_m_x** **spc_m_y** **spc_m_z**	Seatbelt axial force Seatbelt length Slipring pull-through Retractor force Retractor pull-out SPC X force (at node) SPC Y ditto SPC Z ditto SPC X moment (at node) SPC Y ditto SPC Z ditto
User-defined components	**unos** *n* **usss** *n* **ubms** *n*	Nodal scalar component #n So/Sh scalar component #n	Simple formulae may reference other user-defined components as well as the standard components above.	

		Beam scalar component #n	
Material properties valid for Parts and part-based elements	**dens** **ymod** **prat** **ystrs** **fstrn**	Material density Young's modulus Poisson's ratio Yield stress Failure strain	These are calculated by PRIMER and written to the .ZTF file, so they will only be available if a ZTF file has been read. (A binout file is not required.) Not all properties are calculable for all material types, and -1.0 will be returned where values cannot be computed.

Applicability of the "Standard" data components in the table above

The data components above are all implicitly available for nodes.

- Where the data is nodally-derived (eg current X coordinate **cx**) then the value is used directly.
- Where the data is element-derived (for example beam axial force **bfx** , or contact stress **csn**) then the nodal value will be the average of all elements of the relevant type meeting at that node.

Elements may only use components of their "native" type, or nodally-derived data.

- Where an element data component matches the element type (eg X stress **sxx** for shells, or YY bending moment **bmyy** for beams) then it is used directly.
- Where an element references a nodally-derived data component (eg X displacement **dx**) the value used will be the average of all nodes on that element.
- Where an element references an element-derived data component of a different type (eg a shell element refers to beam axial force **bfx**) then the result will be zero.

Applicability of programme settings: Integration point, frame of reference, etc.

- Data is extracted "per integration point" for user-defined components, and it is not currently possible to specify data from some explicit integration point within a Simple Formula, meaning that you cannot use this method to assemble data from multiple integration points in an element.

- The current global / local / cylindrical "frame of reference" (see REF_FRAME... Choosing the frame of reference) does **not apply** when a Simple Formula value is calculated from raw tensor components (eg Sxx) for solids and shells. The unqualified acronym (Sxx, Syz, ...) always extracts global data, and the acronym with an "l" suffix (eg Sxxl, Sxzl) always extracts element local data.

  However if you build a tensor Simple Formula (6 formulae) for solids or shells then the current "Frame of reference" **is** applied when it is **used** . In other words tensor Simple Formulae are created as "raw data", exactly like analysis data read from disk, and are subject to the same transformations when used. It is necessary to work this way in order to prevent "double transformations". (Scalar and vector user-defined components are not transformed.)

Maths functions supported

General functions	**`sqrt(x)`** **`log(x)`** (natural log) **`log10(x)`** (log base 10) **`exp(x)`** (e to the x)	**`mod(x,y)`** (modulo x / y) **`max(x1,x2)`** **`min(x1,x2)`** **`sign(x,y)`** (sign of y transferred to |x|)	**`abs(x)`** **`int(x)`** (truncate to integer) **`nint(x)`** (nearest integer) **`ceil(x)`** **`floor(x)`**
Trigonometric functions (use radians, not degrees)	**`sin(x)`** **`cos(x)`** **`tan(x)`**	**`sec(x)`** **`csc(x)`** **`cot(x)`**	**`asin(x)`** **`acos(x)`** **`atan(x)`** **`atan2(x,y)`**
Hyperbolic functions	**`sinh(x)`** **`cosh(x)`** **`tanh(x)`**		**`asinh(x)`** **`acosh(x)`** **`atanh(x)`**

Arithmetic types used

ALL calculations in simple formulae are processed using double precision floating point arithmetic. This includes constants, and means that the "integer arithmetic" conventions of languages such as Fortran or C, and their associated truncation, do not apply.

> For example the expression **`5 / 2`** is converted to **`5.0 / 2.0`** , and so returns the result **`2.5`**

Also the "integer" functions above (**`int`**, **`nint`**, **`ceil`** , **`floor`**) return integer values, but expressed as floating results.

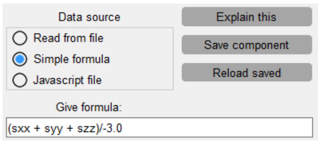
> For example **`nint(3.141592)`** evaluates to **`3.0`**

Examples of simple formulae

`temp - 273.15`	Convert temperatures from Kelvin to Celsius
`(sxx + syy + szz) / -3.0`	Standard calculation for average stress
`1 / sqrt(2) * sqrt((sxx-syy)**2 + (syy-szz)**2 + (szz-sxx)**2 + 6*(sxy**2 + syz**2 + szx**2))`	Standard calculation for von Mises stress
`sxx*exx + syy*eyy + szz*szz + 2.0*sxy*exy + 2.0*syz*eyz + 2.0*szx*ezx`	An estimate of strain energy density in a fully plastic section

Note that the "built in" data components described above can be referred to as simple acronyms, whereas maths functions require their arguments to be placed inside brackets.

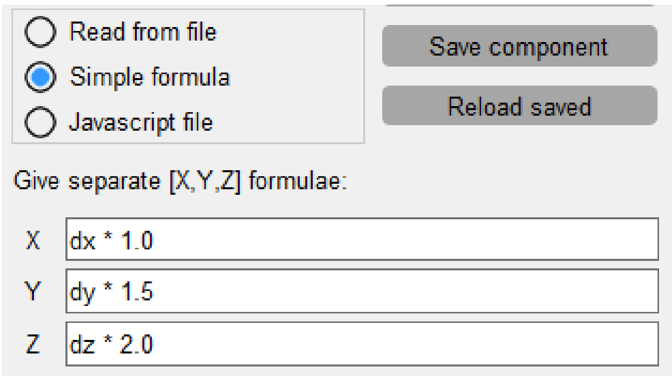
Defining a scalar formula



In the case of a scalar component only a single formula is used.

In the example here the user is calculating pressure at nodes from stress components.

Defining a vector formula



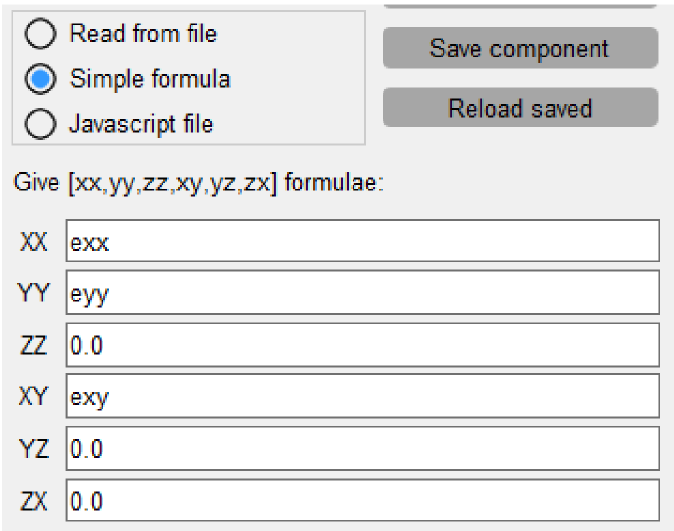
In the case of a vector component three formulae are used, one for each of the [x,y,z] components. These are three wholly separate formulae which are evaluated independently.

In the example here the user is applying factors to displacements at nodes.

Note that each row must have a formula defined for the component to be valid.

Defining a tensor formula

In the case of a tensor component six formulae are used, one for each of the [xx,yy,zz,xy,yz,zx] components. These are six wholly separate formulae which are evaluated independently.



In the example here the user is extracting only the in-plane (2D) components of strain for an element.

As with vectors each row must have a formula defined for the component to be valid even if, as here, some row values are set explicitly to zero.

Further notes on Simple Formulae

Handling errors and missing data

If the expression being evaluated is grammatically legal, but not valid in a given context, then zero is returned. Typical examples are:

- If a formula references a built-in data component that is not present in the model database.
- If a mathematically illegal operation is attempted, such as divide by zero or the square root of a negative number.

Evaluation of element quantities at nodes

If an expression refers to an element data component, for example **sxx** , but is being evaluated for a node, then the relevant value is averaged from all legal elements at that node subject to the current averaging rules in force.

This can result in elements of more than one type being used: in this example component **sxx** , X stress, can be present in solids, shells and thick shells, so if multiple element types use the node then the value will contain contributions from all of them.

In general you should avoid contouring element quanties evaluated at nodes, since the effect is to smear out results and hence miss peak values at element centres. The result is identical to "low resolution" contouring of element data for scalar components.

Evaluation of nodal quantities at elements

If an expression refers to a nodal quantity, for example **cx** current X coordinate, but is being evaluated for an element, then the average value from all the nodes on that element is used.

Limitations of simple formulae

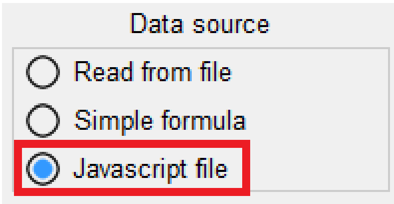
Simple formulae are powerful and very easy to use, but they do have limitations:

- They only operate on the current node or element, at the current surface or layer if relevant.
- They do not permit conditional expressions: you cannot use "if(...)" expression syntax

9.16.6. JavaScript File

JavaScript file

The third way of creating user-defined data components is via an externally defined JavaScript file. Each node or element runs a short JavaScript, the result of which becomes its value.



You can use the JavaScript API Reference Manual to access the full range of "built-in" data components, in the same way that you can with the Simple Formula option above. You can also access other element properties in this way via further data functions.

Using these capabilities, expressions can be evaluated in real-time outside D3PLOT, possibly with reference to external data, and then fed back in for plotting.

The JavaScript File method is limited in that it can only work on the current node or element, however it does provide a bit more functionality than Simple Formulae in that conditional ("if") statements can be used. This makes it possible, for example, to generate results from an internal lookup table.

For more complex processing it is recommended that you consider generating UBIN components using the main JavaScript API, see "User Defined Binary" (UBIN) Components Generated from the JavaScript API below for more information.

How the JavaScript user components work

- You create a JavaScript file (by convention ".js") outside D3PLOT using standard expressions, and also references to built-in data components.

- This is defined as the evaluation method for a user-defined component, currently limited to scalar components.
- It is test compiled to check for syntax errors, and if it is OK the compiled script is stored as the evaluation method for this component
- Thereafter nodes or elements run the compiled script on demand to obtain results.

The value used is the return value of the JavaScript function. By convention this is the result of the final executable statement.

The (generally small) JavaScript function is run for every element or node that requires data evaluation, so for a large model there may be some delay while all the values are computed.

Using built-in data components

The full range of data components used for Simple Formulae is available, with the following differences:

- Each component is a function in JavaScript, not a variable. Therefore it uses the acronym in the Simple Formula table above followed by `()` . For example:

  `sxx` , X stress, is referred to as `sxx()` inside Javas
  Script. Note that the function has no arguments.

- Variable components above, e.g. "extra" values such as `sox` , pass the variable as the function argument. For example:

  `sox 3` , Solid Extra component #3, is referred to as `sox(3)` inside JavaScript.

Using maths functions

Maths functions in JavaScript are supplied by a "Math" class built into the language. For example:

`Math.sin(x)` evaluates sine (x)
`Math.sqrt(y)` evaluates the square root of (y)

Extra functions provided for JavaScript

At present these are limited to:

`label()`	Returns the label of the item (eg node or element) currently being evaluated
`pid()`	Returns the part number of the element currently being evaluated. Zero is returned if called for an item that does not reference a part.

`print(args)` Will write <args> to the controlling terminal.

Examples of JavaScript function

This function evaluates von Mises stress.

Copy Code
JavaScript

```
var i = (sxx() - syy());
var j = (syy() - szz());
var k = (szz() - sxx());

var l = Math.sqrt(i*i + j*j + k*k + 6 * (sxy()*sxy() + syz()*syz()
+ szx()*szx()));

var result = 1 / Math.sqrt(2) * l;
```

Here is another example used to handle a "what should I do if I get divide by zero" problem.

Triaxiality is defined as hydrostatic stress (Sxx + Syy + Szz) / 3.0 divided by deviatoric stress (Svon), which can be written (Sxx + Syy + Szz) / (3 * Svon). Obviously the case may arise that the deviatoric stress is zero, giving a "divide by zero" error, and indeed you may want to write some special value in this case, making a simple formula inadequate. A small JavaScript will solve the problem since it permits conditional branching, for example:

Copy Code
JavaScript

```
var a = svon();
var result;

if(a > 0.0) result = (sxx() + syy() + szz()) / (3.0 * a);
```

In this case a potential divide by zero returns 0.0, but you could substitute any value.

These first two examples both cover the case of using a JavaScript function to generate a scalar value. In order to generate a vector or tensor component, you will need to return an array or object, rather than a single numeric result. Here is a third example which demonstrates how to return an array for a vector component.

Copy Code
JavaScript

```
var a = new Array();

a[0] = dx();
a[1] = dy() * 1.5;
a[2] = dz() * 2.0;

var result = a;
```

Those used to Fortran or C programming should note that JavaScript is a very weakly typed language: numeric variables are "numbers", not integers, floats or other specific data types. Expressions are evaluated using double precision floating point arithmetic.

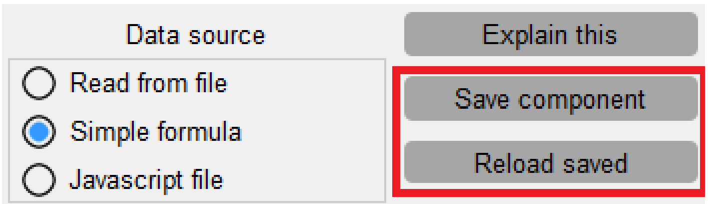
9.16.7. Saving and Reloading User-Defined Components

Saving and reloading user-defined components

All types of user-defined component described above can be saved in a "User Component File" called "d3plot.ucf" using the **Save component** button, which will save the current component in the file for future reuse.

Likewise **Reload saved** will present a menu of components found in the "d3plot.ucf" file, and choosing one will import its settings for the component you are currently editing or creating.

Save component Saving the current component to file



When you **Save** the current component you are asked to choose:

- Which file(s) to save it in.

  OA_INSTALL is your Oasys Suite installation directory.
  OA_HOME is your home directory.
  <cwd> is the current working directory.

  The component will be saved in all files selected for which you have write permission.

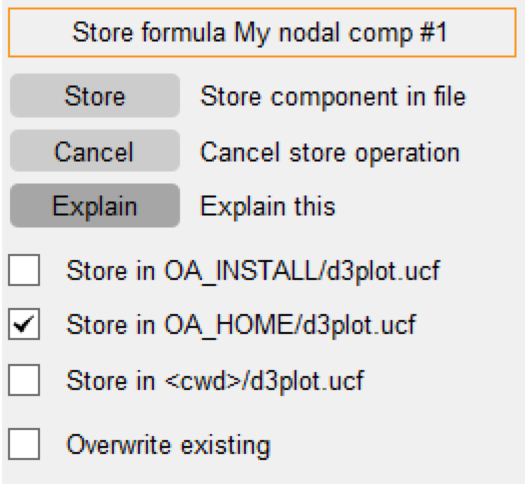
- **Overwrite existing** determines what happens if the name of the component to be stored matches one already in the file.

  If you choose to overwrite then the original component in the file will be overwritten with the new definition.
  Otherwise the current component's name will be added to the file, with its name

modified by adding the suffix #1 (or #2, ...) to make it unique.

Name comparison is not case-sensitive, but does consider embedded white space. For example:

" **`My beam component`** " is treated as being identical to " **`MY BEAM COMPONENT`** "

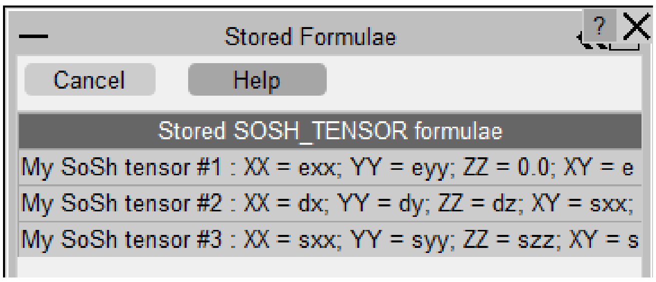


Reload saved Reloading a component from disk to the current definition

When you choose to **Reload saved** the various "d3plot.ucf" files are scanned for components that match the current type, and these are listed in a menu for selection.

In this example the user is using a Simple Formula to define a solid/shell tensor, and there are three components to choose from.

Note that the component "My SoSh tensor #1" has been stored multiple times, and, because **Overwrite existing** was not used, the subsequent definitions have had the digit on the end incremented to make their names unique. (This is because D3PLOT requires components to have unique names, otherwise its internal parsing of components by name can get confused).



Automatic reloading of user-defined components

When D3PLOT starts it automatically scans any "d3plot.ucf" files and loads any " unique " components, meaning that saved user-defined components automatically become available in a new session.

On start-up it scans the following directories:

OA_INSTALL	Typically where the software is installed, and suitable for system-wide components. You may not have write permission to this directory
OA_HOME	Your home directory. This would be suitable for components you want to use for a range of projects.
<cwd> (Current working directory)	The location of this varies: • On Unix/Linux it will normally be the directory from which you started D3PLOT • On Windows it will be the "start in" directory defined on an icon, unless over-ruled by a command-line "start-in" argument If you double-clicked on a file, or dragged a file onto the D3PLOT icon it will be the directory of that file.

In addition whenever a new model is opened any "d3plot.ucf" file in that model's directory is scanned, and any "unique" components are loaded.

Definition of a "unique" component

A component is deemed to be "unique" ***if*** :

- Its name does not match any currently stored component (including "built-in" and UBIN components).

  As explained above name matching is not case sensitive, but does consider embedded white space.

and

- Its data row(s) do not match stored data rows(s)

  FILE and **JAVA** script filename comparisons are made verbatim.

  FORM ulae matching ignores both case and embedded white space. (i.e. the formulae are compressed to remove any white space before testing for matches)

If a component read from file is found not to be unique, i.e. it has already been read and stored, then it is ignored. In this way multiple reads of the same file, or the presence of a formula in more than one file, will not result in multiple definitions being read and stored.

If a component name matches, but its content (data rows) does not, then it is read in and its name has "#1", "#2", etc. added to make it unique.

The Format of the "d3plot.ucf" file

Components are saved in the generic format:

<table>
<tr><td rowspan="3"><Type>
<Source>
<Name></td><td><Type> is one of</td><td>NODE_SCALAR
SOSH_SCALAR
BEAM_SCALAR
OTHR_SCALAR</td><td>NODE_VECTOR
BEAM_VECTOR
OTHR_VECTOR</td><td>SOSH_TENSOR
OTHR_TENSOR</td></tr>
<tr><td><Source> is one of</td><td>FILE
FORM (ula)
JAVA (script)</td><td></td><td></td></tr>
<tr><td><Name></td><td colspan="3">Is a user-defined name up to 30 characters long</td></tr>
<tr><td rowspan="3"><Definition row #1>
: :
(Up to five further rows)</td><td>For FILE a single row</td><td colspan="3">A filename up to 256 characters long</td></tr>
<tr><td>For FORM 1, 3 or 6 formulae. Each on a new line, and each up to 256 characters long.</td><td>xxxx_SCALAR
xxxx_VECTOR
xxxx_TENSOR</td><td>one formula
three formulae
six formulae</td><td><scalar>
<x, y, z>
<xx,yy,zz,xy,yz,zx></td></tr>
<tr><td>For JAVA script a single row</td><td colspan="3">A filename up to 256 characters long</td></tr>
</table>

The file is free-format ASCII (text) and is not case sensitive.
Each row must be on a single line, up to 256 characters long.
Blank lines are ignored.
Lines starting with "%", "$" or "#" in column 1 are treated as comments, and are also ignored.

The only exception is that on Unix/Linux systems filenames ***are*** case-sensitive and they will be stored exactly as they have been defined.

The file may be hand-edited using a normal text editor. It will also be updated by the **Save** and **Reload** options described above.

Here is an example of a typical file

```
$ Example file, created for user manual 1/11/2007

NODE_VECTOR FILE Example of nodal vector file
i:\demos\example1.dat
```

```
NODE_SCALAR FORM Example of scalar node form
sqrt(dx + dy + dz)

BEAM_VECTOR FORM Example of vector beam form
fx * eax
fy * eyz
fz * ezx

SOSH_TENSOR FORM Example of elem tensor form
sxx * exx
Syy * Eyy
sxx * ezz
sxy * exy * 0.5
syz * eyz * 0.5
szx * ezx * 0.5

BEAM_SCALAR JAVA Example of beam scalar Java
/local/demos/javascript/beam_scalar.js
```

9.16.8. Using User-Defined Data Components

Using user-defined data components

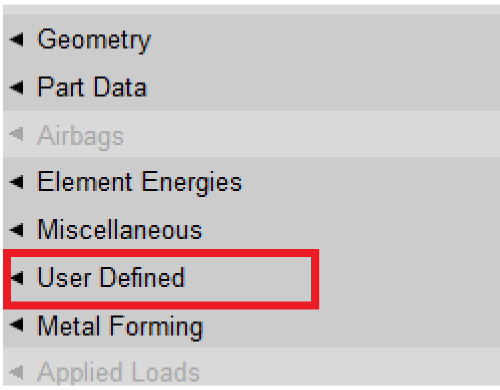
Once they have been created user-defined components become available for use in all contexts where "built-in" components may be used:

- Data plotting
- WRITE and XY_PLOT output

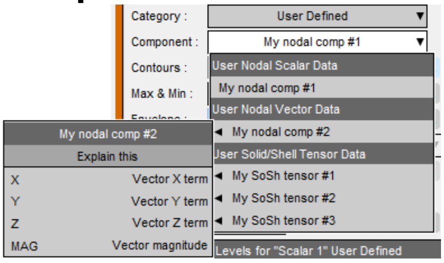
In data plotting

In this context, the **User Defined** category will become available from the relevant tabs in the Data menu.

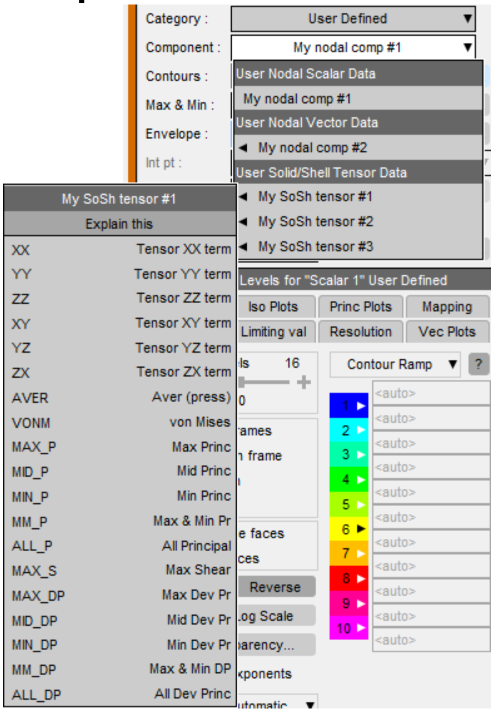
For Scalar data plotting, each data type is listed separately. Scalar components can be selected directly, and the individual sub-components of the vector/tensor cases may be selected from popup sub-menus.



Selecting Nodal Vector sub-components

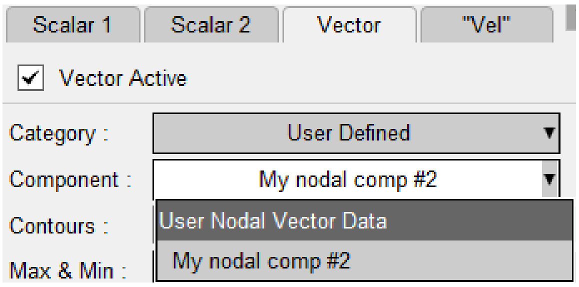


Selecting Element tensor sub-components



Note: The von Mises calculation in this context assumes stress, not strain.
For an explanation see [Computing equivalent strain values](#) .

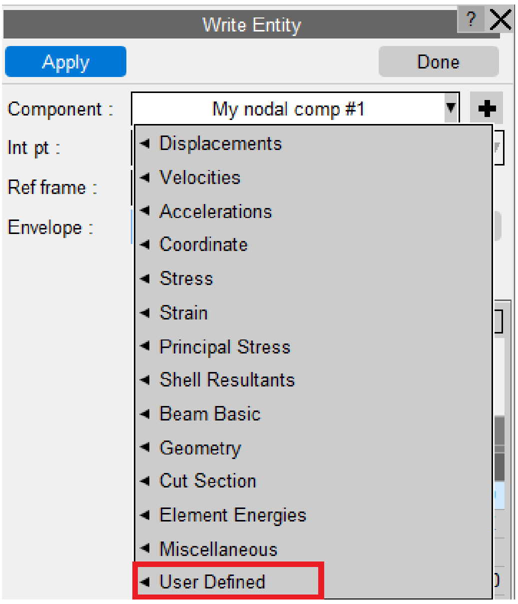
For Vector plots, any Nodal Vector user-defined components are added to the list of components available for vector mode (arrow) plotting. In this context, the vector component as a whole is selected, as all three terms are used to generate the image.



In WRITE and XY DATA panels

In these two contexts the **User Defined** category will become available in exactly the same way as described above, if the relevant user-defined components have been defined.

The sub-menus give access to the top level user-defined components, and below that popup menus give access to the vector and tensor sub-components, again exactly as above.



9.16.9. "User Defined Binary" (UBIN) Components Generated from the JavaScript API

"User Defined Binary" (UBIN) components generated from the JavaScript API

There is a fourth class of user-defined data component that shares all the attributes of those described above, which is used with the JavaScript API (described in The JavaScript Interface , and not to be confused with the "JavaScript File" method above).

UBIN components are far more powerful than the methods above, since data can be extracted for any node or element, from any state and at any integration point; and it can be written anywhere in the same way. In addition, the JavaScript API allows data to be read from and written to external disk files, so UBIN components can be created from any mix of internal and external data using algorithms of arbitrary complexity.

UBIN components are used in the same way as the three types above except that:

- UBIN components can only be created and manipulated via the JavaScript API.
- They are cached on disk in <jobname>.ubd files alonside the normal database files, meaning that they persist across D3PLOT sessions.
- Their data is stored in compact binary form, meaning that access is quick but <jobname>.ubd files are not hand-editable.

Once a UBIN component has been created by running a JavaScript, it will appear in the lists and menus of user-defined components, and can be used in exactly the same was as any of the other user-defined components described above.

The creation and management of UBIN components is described in more detail in the JavaScript Interface Appendix .

9.17. Annotations

The D3PLOT Annotations menu is launched from the Tools menu and can also be found using Quick Find by typing “annotations”, "bookmarks" or “snapshots”. In the Annotations menu you can create annotations composed of titles, descriptions, markers applied to the models on a page, and a page snapshot that can be reloaded later. Annotations can be exported to D3PLOT Viewer.

Note: the crash simulation images in this section are for illustrative purposes only and do not reflect on the actual performance of any vehicle.

- Create and Edit Annotations
- Playback Annotations
- Saving and Retrieving Annotations
- Bookmarks
- Exporting Annotations

9.17.1. Create and Edit Annotations

Creating a Single Annotation

To create an annotation:

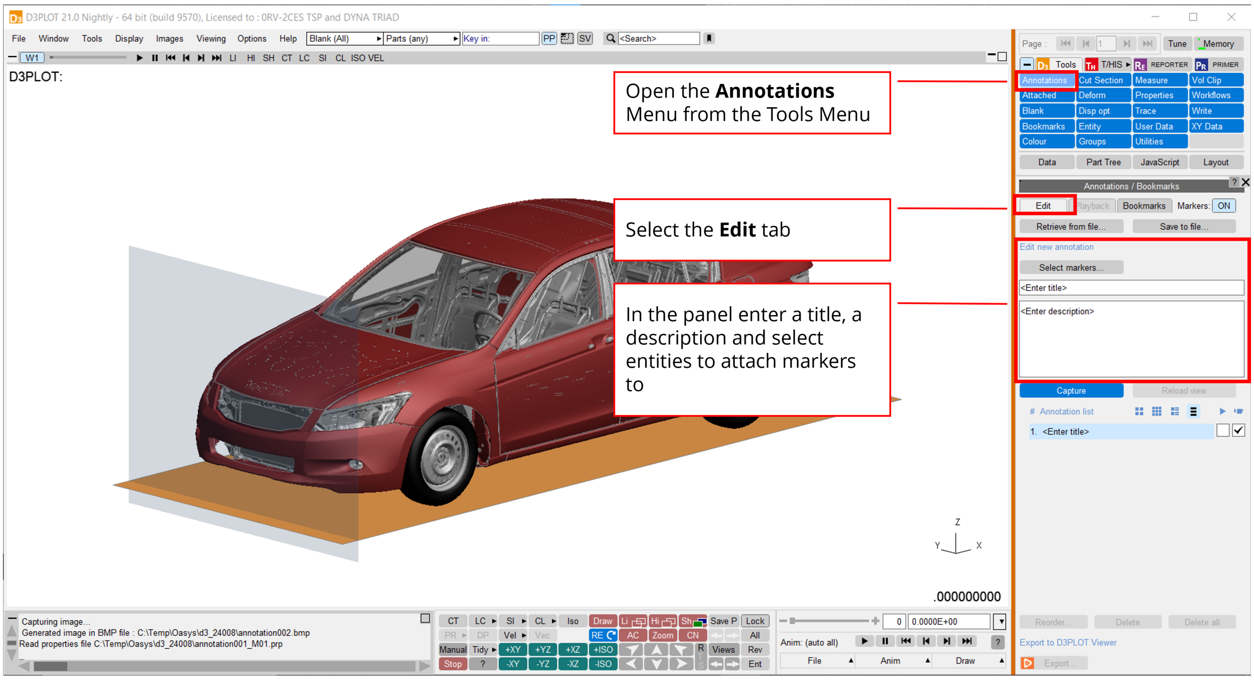
1. Open the Annotations menu from the Tools menu and ensure the **Edit** tab is selected.
2. Select one or more markers to display with the annotation by clicking **Select markers...** and using the Object Menu to select entities to attach markers to.
3. Enter an optional annotation title in the title box. If no title is entered, the “<Enter annotation title>” text will be removed and the title field will be blank for the annotation.
4. Enter an optional description in the description box. This is a multi-line textbox, so it can accept detailed descriptions. If no description is entered, the “<Enter annotation description>” text will be removed and the description field will be blank for the annotation.
5. Finally, click **Capture** to create the annotation. This saves a settings file and properties files for the current page to a temporary directory along with storing the markers, title and description.

The new annotation will be added to the **Annotation list** (below the **Capture** button). Each annotation in this list has two further checkbox options:

1. **Autoplay** [ ]
   If ticked, the annotation will be automatically animated when reloaded
2. **Export as animation** [ ]
   If ticked, if the list of annotations is exported to D3PLOT Viewer, the annotation will be exported as an animation rather than a single frame

The annotation list can be displayed in four different modes, selected by using the icons above the list:

1. **List mode** [ ] displays the title, **Autoplay** and **Export as animation** checkboxes
2. **Thumbnail list mode** [ ] displays a small thumbnail, the title and the **Autoplay** and **Export as animation** checkboxes
3. **Small thumbnails mode** [ ] displays the title and a small thumbnail
4. **Large thumbnails mode** [ ] displays the title and a large thumbnail

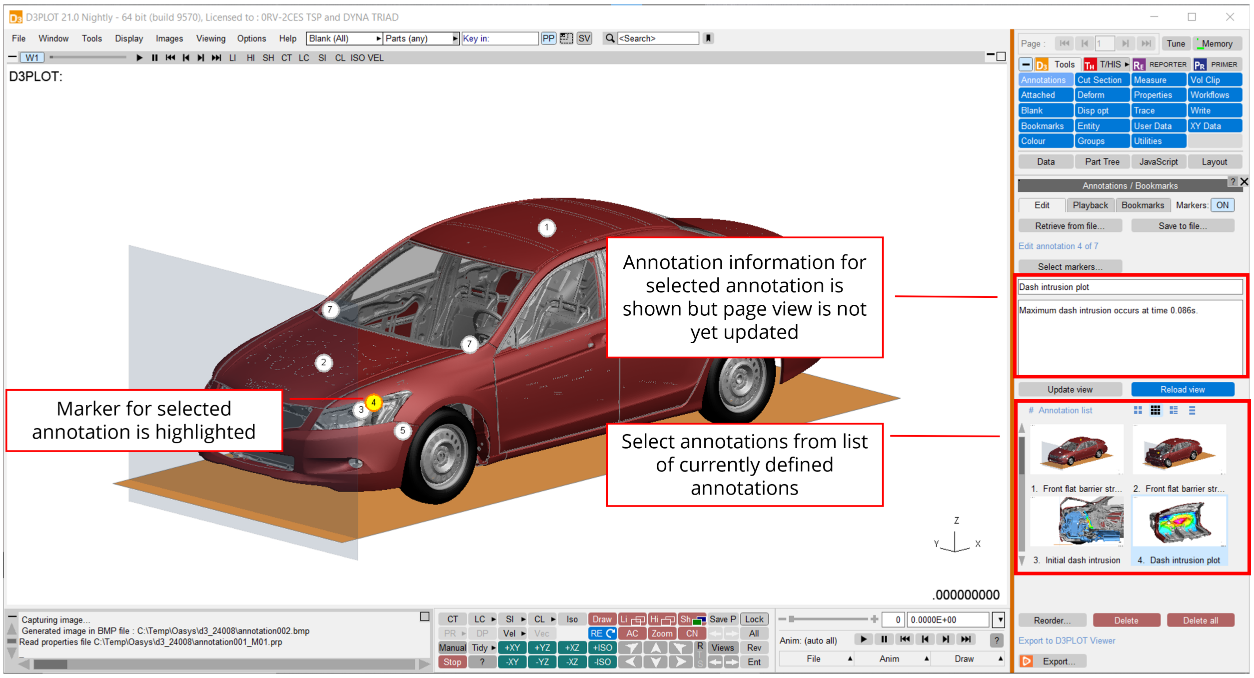


Editing an Annotation

To edit an existing annotation, first select the annotation in the **Annotation list**. This will show the title and description for the selected annotation in those textboxes and will highlight the markers for the selected annotation. The view on the page will not be automatically updated. To reload the annotation page view, click **Reload view**.

The title and description can be updated simply by editing the text in the corresponding textbox. There is no need to click anything else – the edits are automatically stored for the annotation. Likewise, the markers can be redefined by clicking **Select markers...** and selecting new entities to attach markers to (the existing markers will be overwritten).

To update the page view, make any changes to what is shown in the graphics window – change the view, the data component, magnify displacements, etc. – and then click **Update view**, which will save a new settings file and properties files to the temporary area.

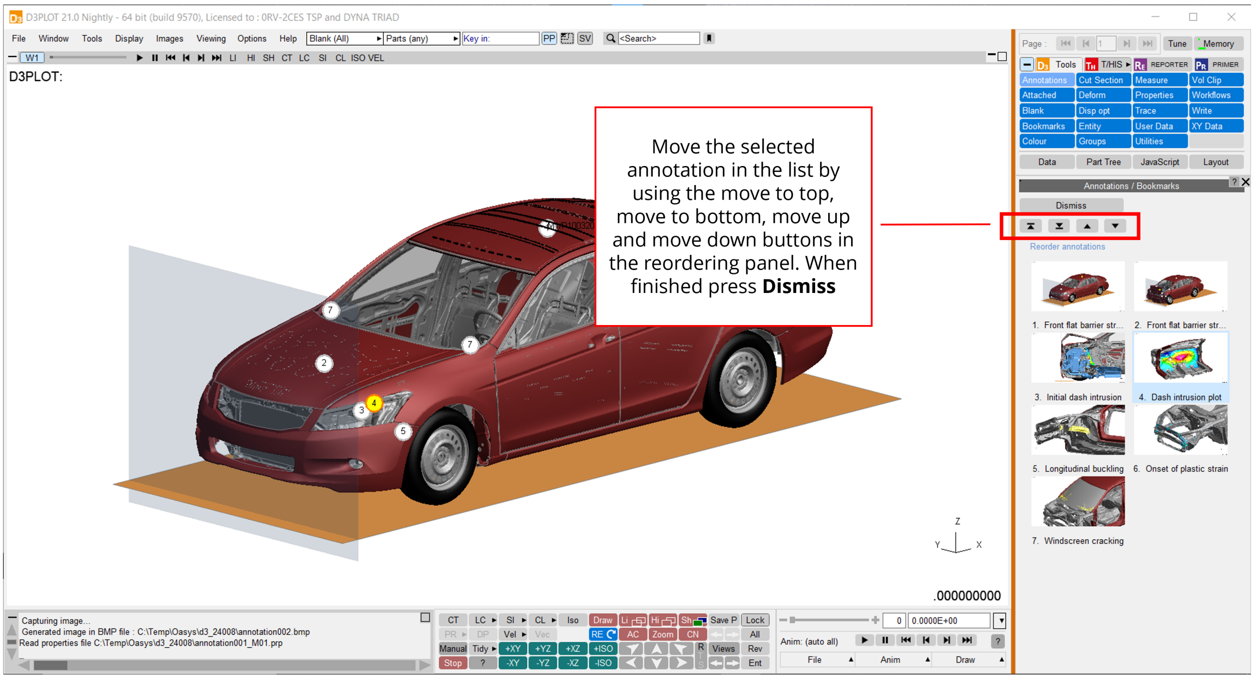


Reordering

To reorder annotations, click **Reorder...** (towards the bottom of the Annotations menu). This will bring up a new menu.

To move an annotation up or down the list, first select it in the list then use the arrow buttons at the top of the menu to move it to the top, the bottom or up or down by one place.

Once the annotations are in the order you want, return to the main Annotations Menu by clicking **Dismiss**.



Deleting Annotations

The selected annotation can be deleted immediately by clicking **Delete** (towards the bottom of the Annotations menu). There is no confirmation: the annotation is deleted immediately.

To delete all the annotations in the list, click **Delete all**. In this case, you will be prompted to confirm the operation before the annotations are deleted.

9.17.2. Playback Annotations

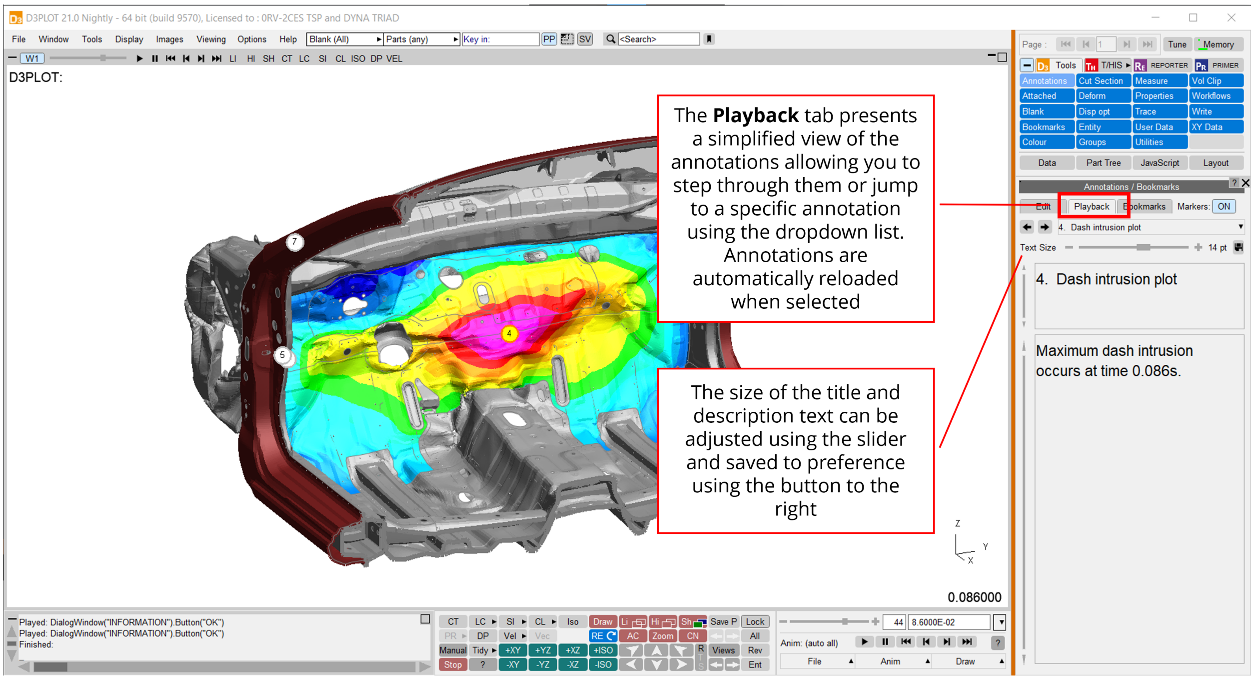
To playback annotations, for example when presenting them, you can select the **Playback** tab at the top of the Annotations menu. This displays a simplified view of the annotations information – just the title and description of the currently selected annotation along with buttons to navigate through the annotations list.

You can step forwards and backwards through the annotations list using the left and right arrow buttons.

You can jump to a specific annotation using the dropdown list.

When you selected a different annotation, the view will automatically reload to show the view for the selected annotation. This is different from the **Edit** tab, where it is necessary to click **Reload view** to reload the annotation view.

You can adjust the size of the title and description text by using the slider at the top of the tab.



9.17.3. Saving and Retrieving Annotations

Saving Annotations

You can save a list of annotations by clicking **Save to file...** (towards the top of the Annotations menu) and entering a filename, which will save a file with a `.ann` extension.

The annotations file saves the title, description, list of markers, thumbnail image, settings file, properties files and filenames for the models on the page. It does not save any of the model files, so these must remain available for the annotations to be reloaded.

Retrieving Annotations

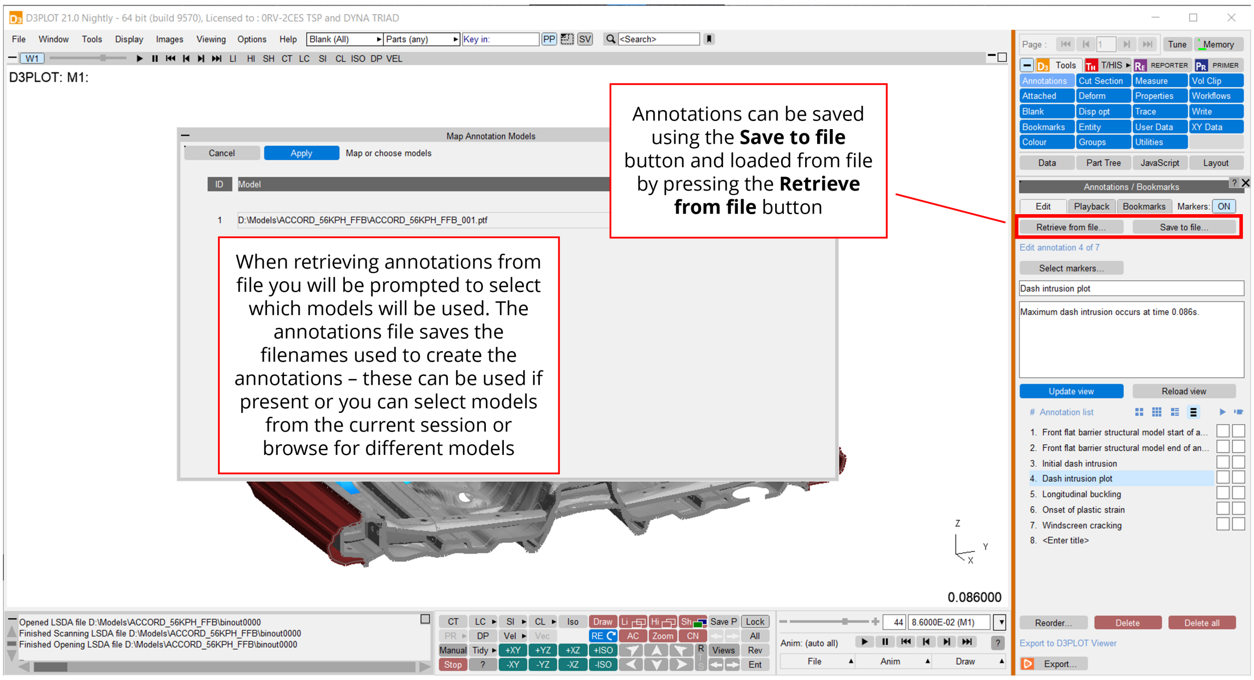
You can retrieve a list of annotations by clicking **Retrieve from file...** (towards the top of the Annotations menu) and browsing for an annotations file. This will overwrite any annotations currently in the annotations list.

Once an annotations file has been selected, you will be presented with a list of the models referenced by the annotations and the option of changing these to different models. Alternative models can be selected from a list of the models in the current session using the dropdown, or by browsing for a model using the file selector button.

Once you are happy with the list of models, click **Apply** to reload the annotations file. Alternatively click **Cancel** to return to your D3PLOT session and any annotations you had created in that session.

If the model referenced in the annotations file cannot be found by D3PLOT then the Model row will show “<Undefined><*model filename from annotations file*>” and the **Apply** button will be disabled until all rows point to an available model.

An annotations file can also be reloaded from D3PLOT's **Open Plot File** menu when D3PLOT is first opened, by selecting **Annotations file** from the **Open** dropdown list (annotations files cannot be opened from this menu during a session). Once you have mapped models, D3PLOT will automatically reload the first annotation.


Annotations can be saved using the Save to file button and loaded from file by pressing the Retrieve from file button
When retrieving annotations from file you will be prompted to select which models will be used. The annotations file saves the filenames used to create the annotations – these can be used if present or you can select models from the current session or browse for different models
Map Annotation Models
Cancel
Apply
Map or choose models
Retrieve from file...
Save to file...
Dash intrusion plot
Maximum dash intrusion occurs at time 0.086s.
0.086000

9.17.4. Bookmarks

Bookmarks

The **Bookmarks** tab offers a quick and easy interface to annotations for situations when adding markers, titles and descriptions is not required. The **Bookmarks** tab can be opened directly from the **Tools** menu. The **Bookmarks** tab is a different view of the list of annotations, so when you capture a bookmark it is added to the list of annotations and vice-versa.

To capture a bookmark, click **Capture**. The current page view will be added to the end of the list as a new bookmark.

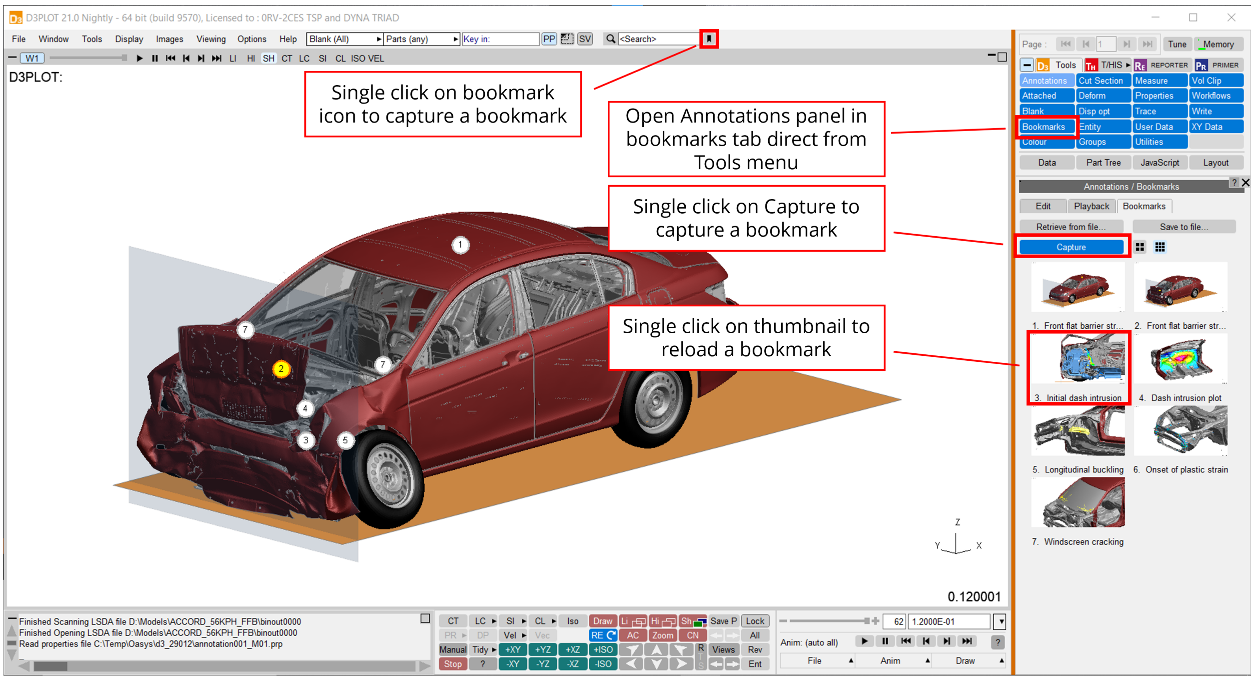
To reload a bookmark simply click on its thumbnail and it will be reloaded immediately.

The size of the thumbnails in the list can be changed using the two icons above the list:

1. **Small thumbnails** [ ]
2. **Large thumbnails** [ ]

Bookmarks can also be captured at any time without opening the **Bookmarks** tab by clicking the **bookmark** button [ ] on the top bar.

To edit the bookmarks list – delete unwanted bookmarks, update views, etc. – switch to the **Edit** tab and use the tools in that tab to make your changes.



9.17.5. Exporting Annotations

Annotations can be exported to GLB format for reading into D3PLOT Viewer by clicking **Export** (towards the bottom of the Annotations menu). Current limitations in D3PLOT Viewer export include:

- No support for T/HIS graphs
- Model data is exported for each annotation, so file sizes can quickly become large. File sizes can be minimised by exporting single frames rather than animations and by exporting selected parts or subassemblies of interest, rather than annotating a view of the entire model.

9.18. Workflows

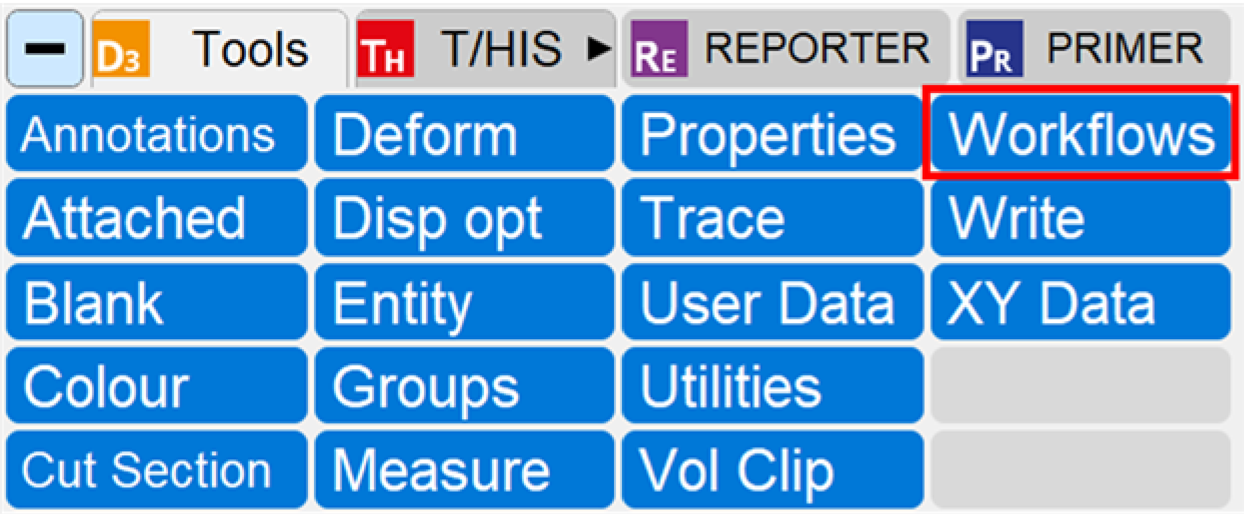
Workflows

The Oasys Suite contains powerful tools and capabilities that can be used to interrogate and debug your analysis results. However...

1. The tools are not always customised for your specific loadcases or tasks
2. You may need to manually perform a number of steps to process your results, which can be time-consuming and prone to error
3. The JavaScript API can be used to create tools to automate your post-processing workflow, but this requires time, resource and knowledge, which is not always available

To address these issues, the Workflows feature provides tools customised for specific loadcases and tasks, built upon the existing capabilities in the Oasys Suite, to make it easier to interrogate and post-process results.

The Workflows framework provides a simple structure to transfer data from PRIMER to the post-processing software. Browse the selection of already-available Workflow Tools, or read more about the Workflows menu in D3PLOT.



9.18.1. Workflows Updates

Workflows 21.1 (released with Oasys 21.1)

Many LS-DYNA pre- and post-processors offer a range of tools that can be used to configure and interrogate LS-DYNA models. However:

1. The basic tools are not always customised for LS-DYNA, or for specific loadcases
2. You may need to perform many manual steps to process your results, which can be time-consuming and prone to error
3. Scripting APIs can be used to create tools to automate tasks, but this requires time, resource and knowledge, which is not always available

To address these issues, we introduced Workflows. Workflows is a powerful framework that provides you with customised tools that work seamlessly from pre-processing through to post-processing, providing results quickly and reliably.

Workflow Tools

Learn how to use the latest Workflow tools here.

Learn about Workflow User Data.

In addition to the tools provided, you can create your own bespoke tools. Please contact us if you have an idea for a tool and would like some help creating it.

Releases

The Workflows tools are constantly being improved and enhanced. If you have any requests for new features or experience issues using Workflows, please contact us and we will aim to address your requests in future releases. This documentation is for 21.1 (released with Oasys 21.1). A record of all releases is listed below.

Workflows Releases

Date	Release	Description	Documentation	Minimum required Oasys Suite version	Download	Changelog
	21.1	• Added support for C-NCAP Far	Workflow Tools 21.1	21.1		

Workflows Releases

		Side Occupant Protection Protocol • Migrated the IIHS and US NCAP automotive library templates from REPORTER to the Workflows framework • Several big fixes				
27-Aug-2024	21.0 W1	21.0 Workflows Update 1 – Comprehensive support for Euro NCAP Virtual Far Side Protocol v1.0	Workflow Tools 21.0 W1	21.0	Oasys 21.0 W1 Workflows Bundle	Oasys 21.0 W1 Changelog
15-May-2024	21.0	First version containing Virtual Testing tools	Workflow Tools 21.0	21.0	Oasys Suite download	Oasys 21.0 Release Notes
08-Nov-2023	20.1	Bug fix release	Workflow Tools 20.1	20.1	No longer available	Oasys 20.1 Release Notes

Workflows Releases

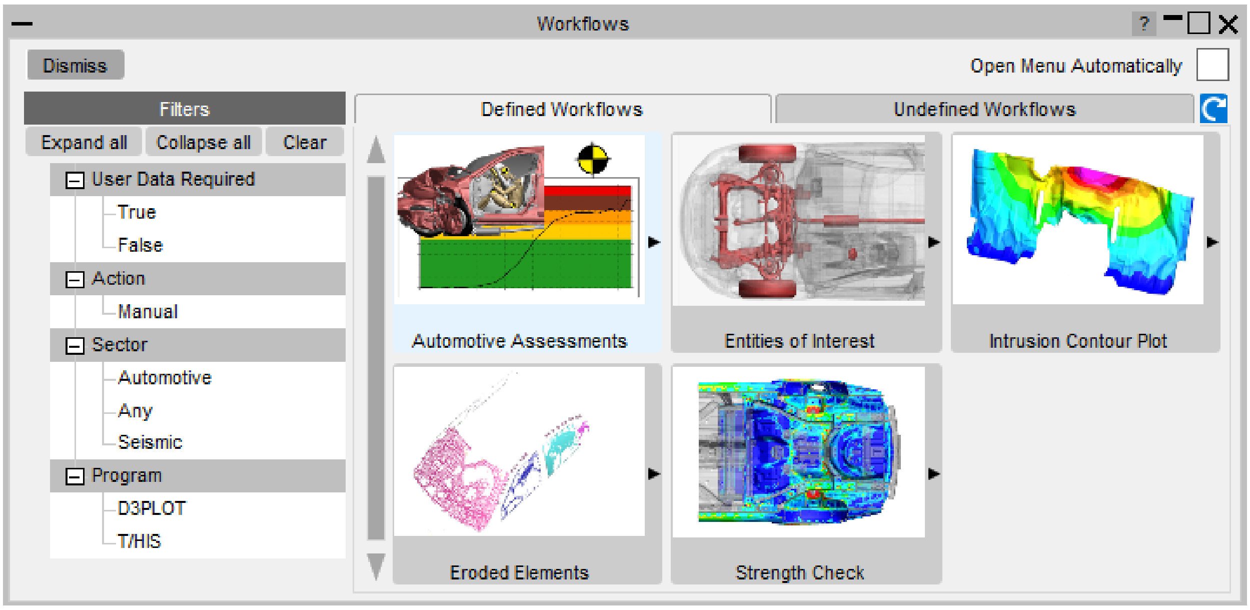
17-Apr-2023	20.0	First release of Workflows	Workflow Tools 20.0	20.0	No longer available	Oasys 20.0 Release Notes

9.18.2. Workflows menu in D3PLOT

Workflows menu in D3PLOT

Tools → Workflows

The Workflows menu allows easy access to all the available workflows. In D3PLOT, you can open the Workflows menu by selecting **Tools → Workflows**. The **Defined Workflows** tab shows all the Workflows that can be run. Selecting any of the Workflows will run the JavaScript defined in the Workflow Definition. Running a Workflow will minimise the Workflows menu and open the script's user-interface. The Workflows menu will maximise again after closing the script.



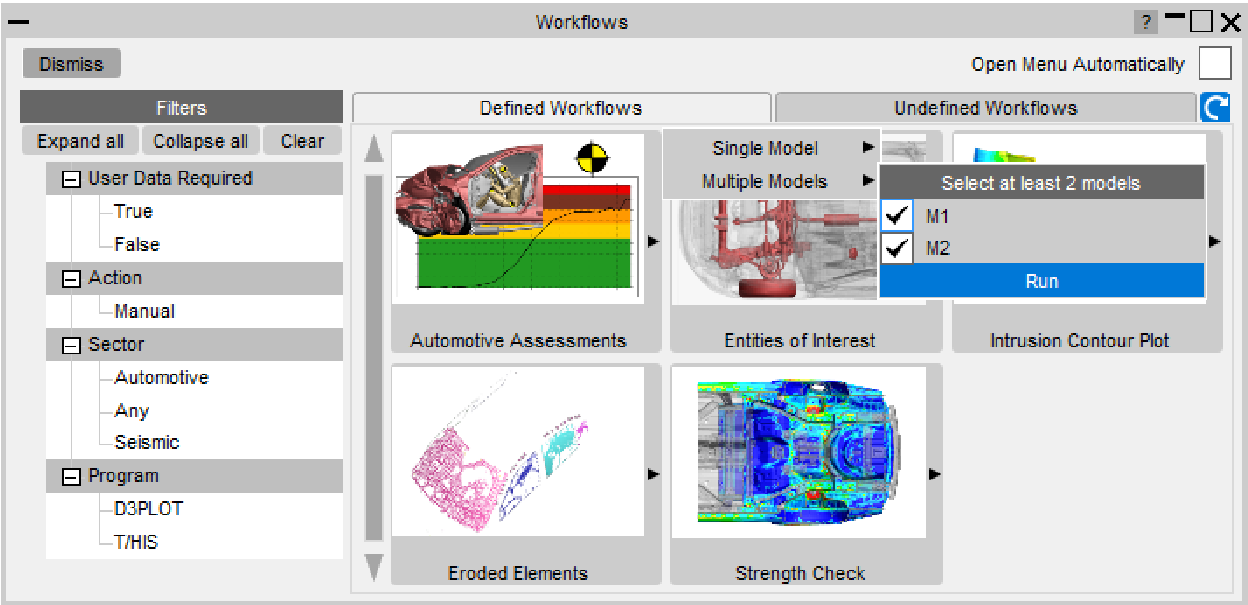
When the Workflows menu is initially opened, it shows all the available Workflows, but you can filter the Workflows by using the **Filters** tree. You can select multiple categories to filter the available Workflows displayed. Individual categories can be selected with single clicks; pressing the **Ctrl** and **Shift** keys while clicking on the tree will do a multi select.

Every Workflow can be tagged with Category and Value pairs, which populate the Filters tree. Selecting multiple Values within a Category will show all the Workflows tagged with any of the selected Values. However, selecting Values across Categories will show workflows which are tagged with all the selected Category/Value.

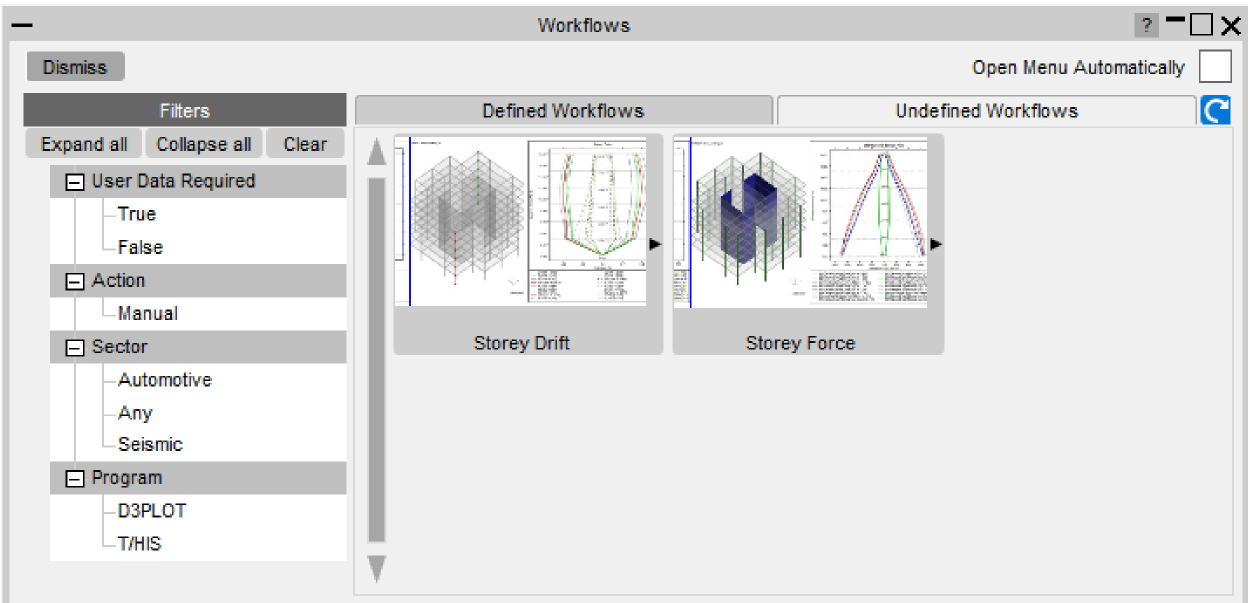
You can always refresh the Workflow user data and available Workflows shown in the menu by using the refresh button in the top-right corner of the panel. This will reread the Workflow data from the files so that the updated data is available. This is useful if you update the Workflow data from a PRIMER session and you want to use the revised user data in your already-open D3PLOT session.

In both D3PLOT and T/HIS, you have the option to automatically open the Workflow menu when reading a model that has associated workflow data, by selecting the **Open Menu Automatically** tick box at the top-right corner of the panel.

If there is more than one model in memory and multiple models have the same Workflow associated with them, the individual Workflow buttons in the **Defined Workflows** tab will have a drop-down option to select for which models you wish to run the Workflow (see example below). You can select a single model or multiple models. The maximum and minimum number of models you can use in a Workflow is defined by the Minimum Multiple Models and Maximum Multiple Models parameters in the Workflow Definition.

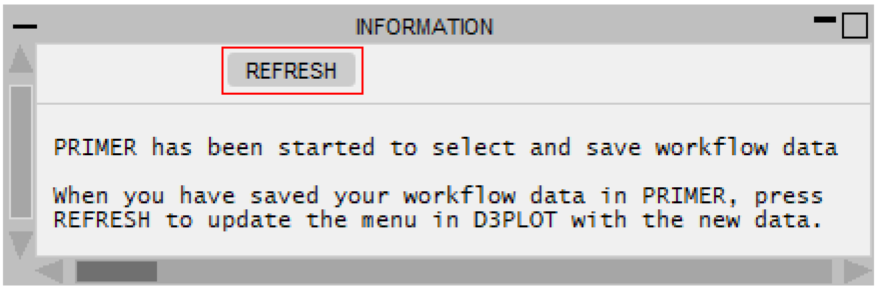


The **Undefined Workflows** tab shows all the workflows that can be run in D3PLOT, but don't have the required user data to run them.



Selecting one of the workflows will open the model in PRIMER and start the workflow so you can select the required data.

In D3PLOT a window will open telling you to press **REFRESH** when you have saved the data in PRIMER. This will update the workflow menu, moving the workflow to the **Defined Workflows** tab so it can be run in D3PLOT.



10. Images

IMAGES

The **IMAGES** option allows static screen images and animations to be captured and also to be read in as background.

"Bitmap" static images and animations are handled using the menu below, whilst for Laser plotting see LASER PLOTTING.

10.1. Creating Static Images and Movies

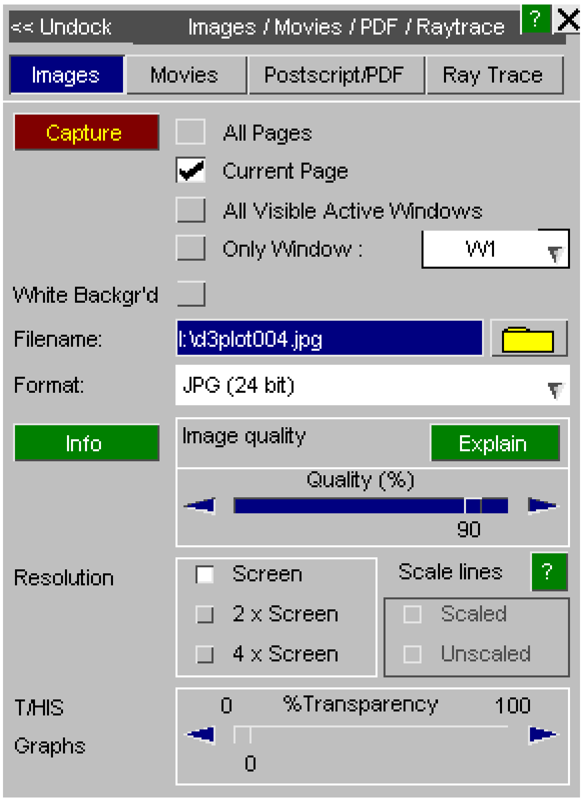
Creating static images and movies

CAPTURE

Captures a single static frame in one of the following formats:

Quick links to:

Movies	Capturing animations
Postscript/PDF	Capturing laser printer and PDF files
Ray-tracing	Generating ray-traced output
D3PLOT Viewer	Capturing 3D models for D3PLOT Viewer



10.1.1. 8-Bit File Formats

8-bit file formats

BMP Uncompressed	Uncompressed 8 bit Microsoft windows bitmap. The approximate size of the file (in bytes) is file size= image width * image height
BMP Compressed	8 bit RLE Microsoft windows bitmap.
PNG	8 bit Portable Network Graphics
GIF	Graphics Interchange Format

10.1.2. 24-Bit File Formats

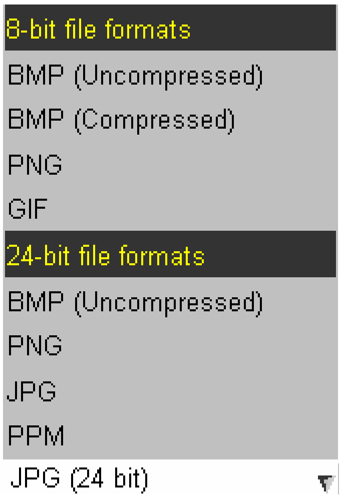
24-bit file formats

BMP	Uncompressed 24 bit Microsoft windows bitmap. The approximate size of the file (in bytes) is file size = 3 * image width *image height
PNG	24 bit Portable Network Graphics
JPG	JPEG (Joint Photographic Experts Group) file

PPM

Uncompressed **P** ortable **P** ix **M** ap. The approximate size of the file (in bytes) is

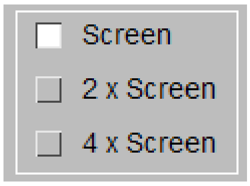
file size = 3 * image width *image height



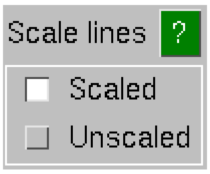
Various `.bmp` formats are available, and there are [Controls](#) for the dithering of the 8 bit-plane variants and palette optimisation .

RESOLUTION

All images can be output at either the screen resolution or at a resolution of either 2 or 4 times the screen resolution.



The widths of lines will appear to get narrower at 2 or 4 times the screen resolution. Scaling can be turned on to make the line widths match those on display.



CAPTURE

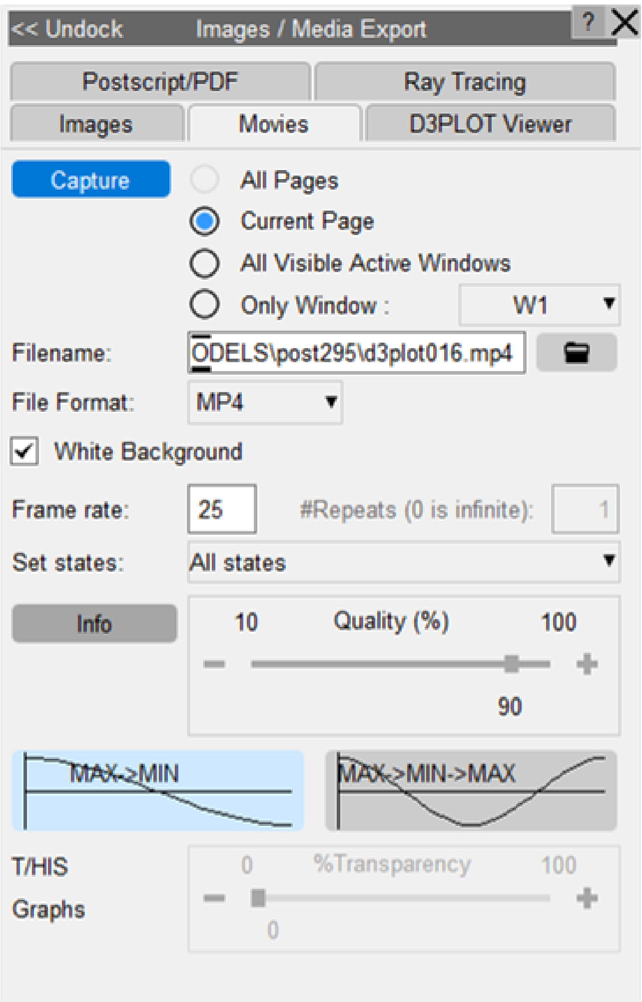
Captures the current animation in one of the following formats:

MP4 `(.mp4)`

GIF `(.gif)`

AVI `(.avi)`

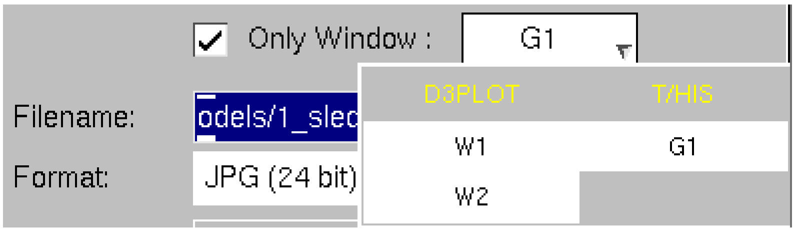
The file format, replay characteristics and frame repeat parameters can be defined.



10.1.3. Multiple Windows

Multiple Windows

When multiple windows are present, any or all of these may be included in the images and animations by using the tabs and dropdown menu. If the T/HIS link is open the menu will have two columns; one for D3PLOT windows and one for T/HIS windows. The captured image will be the size of the window.



10.1.4. White Background

White Background

With this option switched on images will be captured with a white background. Entity labels and screen text will be switched to black. Once the image has been captured the screen will return to its original colours.

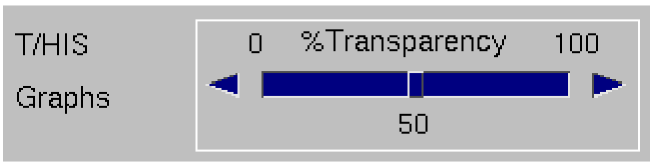
10.1.5. Capturing Composite Images of Linked T/HIS and D3PLOT Windows

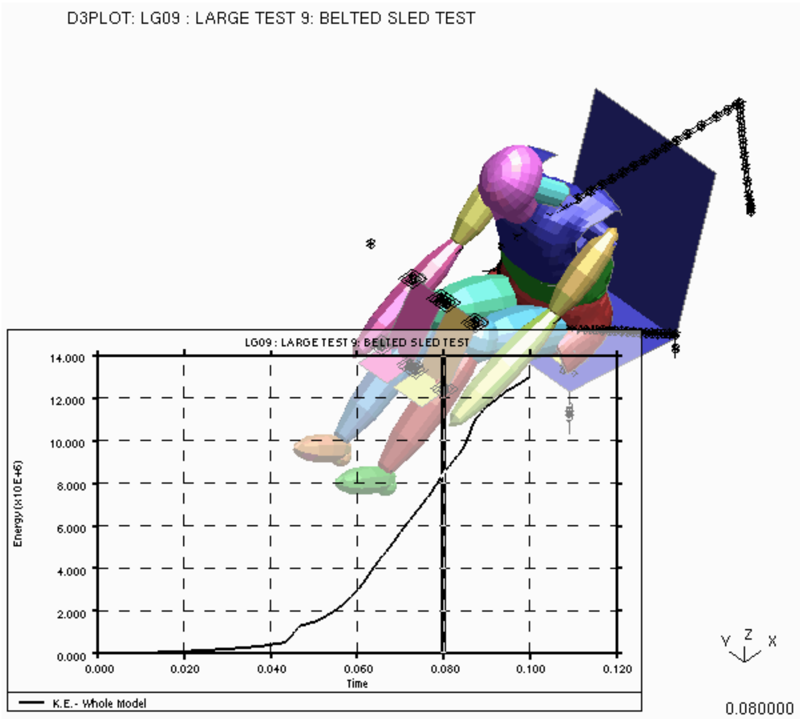
Capturing composite images of Linked T/HIS and D3PLOT Windows

When linked T/HIS is running it is possible to capture a composite image of both D3PLOT and T/HIS windows, and in addition windows may be made partially transparent. This is useful if the T/HIS window is a child of a D3PLOT window so that the underlying D3PLOT image can be seen.

The figure below shows an example of a 50% transparent "docked" T/HIS image overlying the D3PLOT one.

In the undocked or "sibling" cases a composite image will be the size of the rectangular bounding box required to enclose the selected windows.





10.2. Static File Formats Supported

Static file formats supported

JPEG	Joint Photographic Experts Group compressed format. This gives image quality nearly comparable to 24 bit-plane bitmaps, but with a file of < 5% the equivalent size. JPEG format is supported by all common visualisation packages and is recommended for all applications unless image quality is of paramount importance.
8 Bit Compressed BMP :	8 bit runlength encoded (RLE) Microsoft windows bitmap.
8 Bit Uncompressed BMP :	8 bit uncompressed Microsoft windows bitmap. The approximate size of the file is [image width * image height] bytes.
24 Bit Uncompressed BMP:	24 bit uncompressed Microsoft windows bitmap. The approximate filesize is [3 * image width * image height] bytes.
PNG:	24 bit lossless compressed **P** ortable **N** etwork **G** raphics image. PNG offers the similar degree of compression as GIF but has better colour quality.
GIF:	8 bit lossless compressed **G** raphics **I** nterchange **F** ormat.
PPM:	24 bit uncompressed **P** ortable **P** ix **M** ap. The approximate size of the file is [3 * image width *image height] bytes.

10.2.1. Controls on the Quality of 8 Bit-Plane Bitmap Files

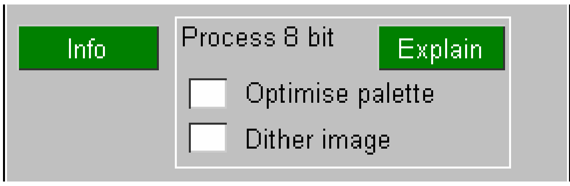
Controls on the quality of 8 bit-plane bitmap files

24 bit BMP files tend to be huge, and the space saved by using a compressed format is attractive. In most cases, JPEG or PNG should be the compressed format of choice.

If an 8 bit format must be used (e.g. **when producing an animated GIF**), there is no avoiding some loss of quality. This is because truncation of a 24 bit format (16 million colours) to 8 bits (256 colours) gives rise to "banding", in which the least significant bits of the original colour definitions are lost. The following options can be used to help alleviate this issue.

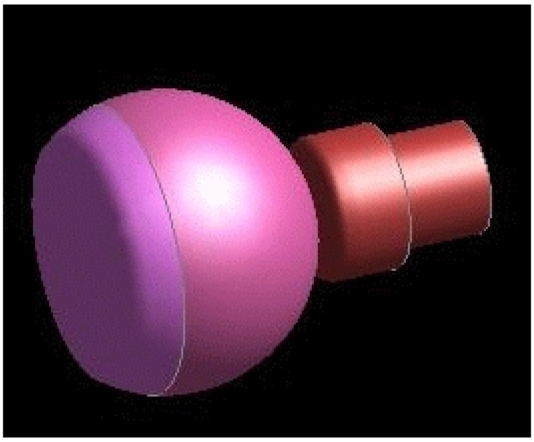
Dither image

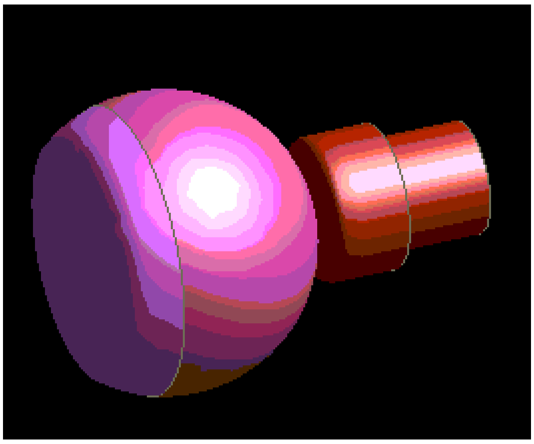
The following sequence of images show how the different levels of dithering affect the quality and file size of a compressed 8 bit image, comparing it with the JPEG equivalent.



For static images there is no advantage in using BMP files over JPEGs or PNGs: BMP files are larger and of inferior quality.

For animated files, we recommend using the MP4 option to produce high-quality output in a widely-supported format. However if you have to revert to an 8 bit AVI format, or if you want to produce an animated GIF, the various permutations of options below will be of interest when trying to obtain the best compromise between image quality and overall file size.





Here is the original 24 bit-plane image, saved as a JPEG file. **Size 5.1kB**

This is the undithered equivalent bitmap image. **Size 7.3kB** .

Note how the discretising affect of mapping onto a limited colour palette has caused "banding" which makes the image almost unusable. However at least the files are small!

"Dithering" is a technique in which an ordered pattern of noise, `xxxxx` in the table below, is added to the least significant bits of a colour value to make it alternate between two adjacent shades.

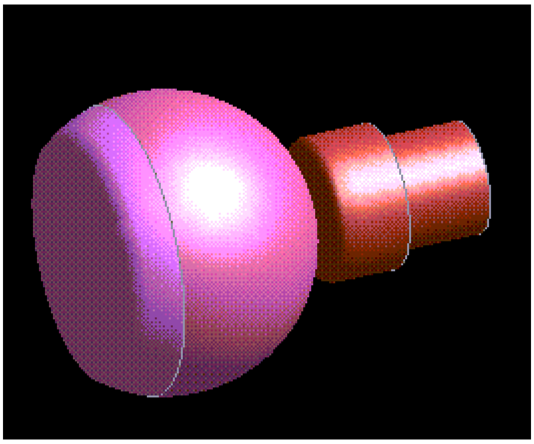
Consider the bits for a single colour in this image that have been truncated from 24 bits (8 bits each of Red, Green and Blue). Truncated bits are shown in lower case.

<table>
<tr><td>Original Red byte <code>001?????</code></td><td>truncates to</td><td><code>001ooooo</code></td></tr>
<tr><td colspan="3">Adding the dither pattern <code>oooxxxxx</code> to the bottom 5 bits of the original byte gives</td></tr>
<tr><td><code>001?????</code> + <code>oooxxxxx</code></td><td>giving either
or</td><td><code>001ooooo</code>
<code>010ooooo</code></td></tr>
</table>

The result may be truncated to `001ooooo` , or the increased to the next shade up `010ooooo` , depending on the trailing bits `?????` and the noise value `xxxxx` at that pixel.

The effect is to produce a composite shade that is somewhere between the two originals.

Here is a dithered version of the image above, **size 8kB** .



Some banding is still visible, but it has been reduced to an acceptable level, and the coarsening of spatial resolution is also evident.

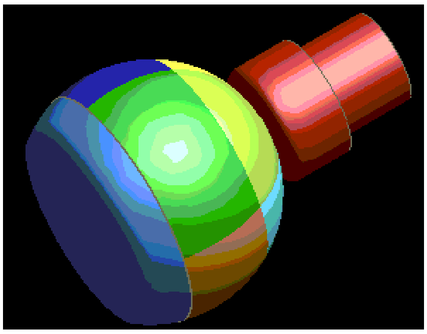
Generally dithering gives the best results for animated 8 bit images

Palette Optimization

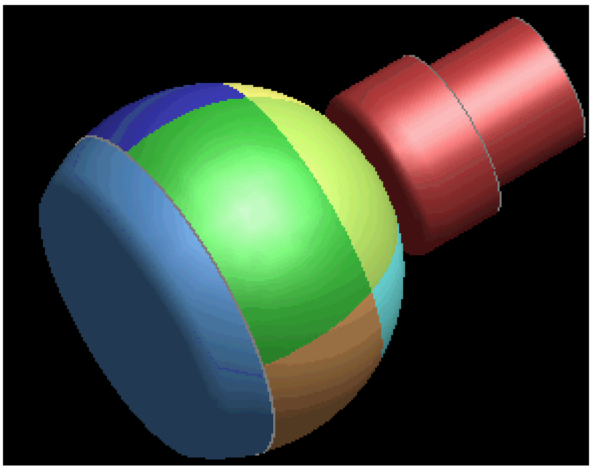
When 8 bit images are produced the 24 bit palette has to be reduced to only 256 colours. To do this the best way is to use Palette Optimization to choose the most representative colours used in the image.

Without Palette Optimization 256 colours can be chosen uniformly along the original 24 bit palette, missing out important colours.

The following figures show the differences in images with Palette Optimization.



This is the original image, saved as a GIF, with no dithering or palette optimization. **Size 6.1kB**

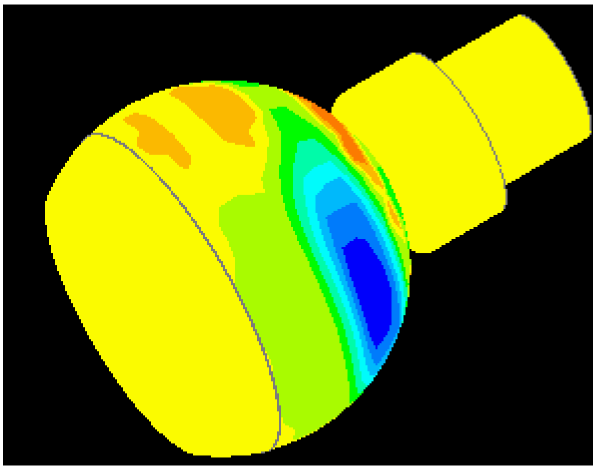


This is the image, saved as a GIF, with palette optimization. **Size 12.6kB**

Note that whilst there are still bands, the coarseness of them has diminished.

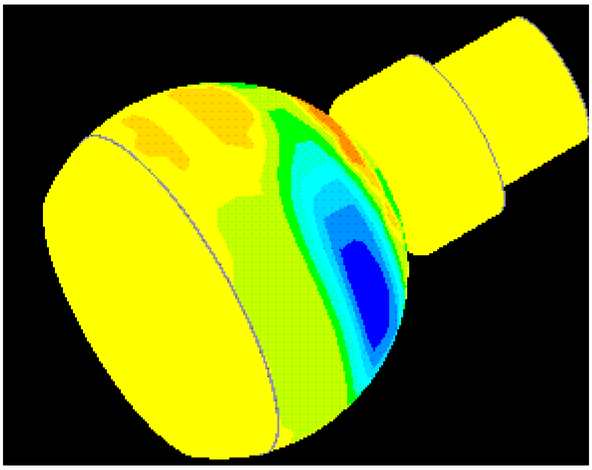
The two images above were created from a shaded image plot in D3PLOT and therefore contained a lot of different colours. The banding is present because the palette has been reduced to the 256 most representative colours in the image.

The following figures are images of a contour plot in D3PLOT.



This is the image, saved as a GIF, with palette optimization. **Size 3.1kB**

Note that due to the smaller number of colours in the image, there is no banding.



This is the image, saved as a GIF, with dithering. **Size 3.7kB**

Dithering is not well suited to images with distinct colours since by its nature it produces colours that are somewhere in between neighbouring colours. This is effective with shaded images, but not with images where there are sharp changes of colour.

10.3. Animation File Formats Supported and their Attributes

Animation file formats supported and their attributes

D3PLOT supports three animation formats:

MP4 (`.mp4`)	**MPEG-4 Part 14** This is a versatile and modern digital multimedia container format, used in D3PLOT for storing video data. This data is encoded using the H.264/MPEG-4 Advanced Video Coding compression format to produce high-quality video content at a reasonable filesize. Image quality and filesize with H.264 encoding are controlled by a number of factors. One of these is the nature of the "GOP" structure (group of pictures). This refers to how "I" frames, containing full information, are mixed with several "P" frames containing only the difference between this and the preceding frame(s). The max GOP size in D3PLOT for MP4 is currently set to 5, such that each I frame is accompanied by at most 4 P frames. In cases where there are significant differences between successive frames, the encoder should be able to detect this and opt to use fewer P frames. Another factor to consider is the bitrate. This can be controlled through the Quality slider. Setting Quality to 100% corresponds to a maximum bitrate of 100Mbps. This is very large and should be more than enough for almost any use case. If the encoder detects that this bitrate is in excess of what it needs to achieve optimal image clarity, it will automatically drop to a smaller value to keep filesize minimal. The H.264 encoder we use requires that both width and height of each video frame be divisible by 2. In D3PLOT, we automatically shrink the window size by 1 pixel in any dimension which does not adhere to this at the time of video capture. MP4 is our recommended choice for widely replayable, high quality video output.
AVI (`.avi`)	**Adapted Video for Internet.** This format acts as a "wrapper" around a sequence of static images, adding information about their content and the replay rate required. In principal any still image format can be wrapped in this way, but in practice commonly available players only support a limited range of formats; and the coder/encoder ("codec") software for higher performance formats tends to proprietary. For example the Indeo, Cinepak and Sorenson encodings are all copyrighted and expensive to obtain. D3PLOT supports MJPG(Motion JPEG) encoding from D3PLOT 9.3 which

	gives great compression, and each frame gets good image quality which is the same as a JPEG. D3PLOT supports also bitmap encoding which, while it does not give good compression or image quality (unless the prohibitively voluminous 24 bit option is used), is at least in the public domain and will guarantee to play back in any package. The dithering options (see **Controls**) that have been added in D3PLOT 8.2 give a reasonable compromise between image quality and file size when 8 bit compressed bitmaps are used.
GIF (`.gif`)	**Graphics Interchange Format.** This format acts as a "wrapper" around a sequence of static GIF images, adding information about their content and the replay rate required. GIF animations will tend to be smaller than AVIs and have the advantage that they can be inserted into Powerpoint as images. This means that unlike AVIs the files do not need to be carried separately with the presentation.

10.3.1. FRAME_RATE Frame Rate when Played Back

FRAME_RATE Frame rate when played back

This option can be used to specify the desired playback speed in frames/second.

The playback speed of an MP4 or AVI movie is encoded into the file along with the length of the movie (seconds). If too high a playback speed is requested then most movie players will skip some frames to ensure that the movie plays for the correct length of time.

The playback speed of a GIF animation is also encoded into the GIF file.

10.3.2. FORMAT AVI File Formats Supported

FORMAT AVI file formats supported

AVI movies can be written using a wide range of file formats. D3PLOT supports the following four, since they have been found to play successfully using 'xanim' on a range of UNIX machines and Microsoft's ActiveMovie on PC's.

MJPG:	Each frame within the movie is stored as an 24 bit JPEG image. This format offers the best combination of quality and size

	and is recommended unless your player will not support it.
8 Bit Compressed :	Each frame within the movie is stored as an 8 bit RLE Microsoft windows bitmap.
8 Bit Uncompressed :	Each frame within the movie is stored as an 8 bit uncompressed Microsoft windows bitmap. The approximate size of the movie is: [#frames * image width * image height] bytes
24 Bit Uncompressed :	Each frame within the movie is stored as an 24 bit uncompressed Microsoft windows bitmap. The approximate size of the movie is: [3 * #frames * image width * image height] bytes

The **Controls** apply equally to the 8 bit formats here in exactly the same way as they do to static images.

10.3.3. QUALITY The Playback Quality of MP4 Files

QUALITY The playback quality of MP4 files

For MP4, Quality controls the average bitrate the encoder should aim for when encoding video data. This is an upper bound and has been set very high, with 100% Quality corresponding to a bitrate of 100Mbps. If the encoder detects that a smaller bitrate will suffice for optimal image quality, it will make this adjustment automatically. It is rare that you will need to interact with this slider for MP4 output.

In D3PLOT the default quality is set to 90%. File size will be broadly equivalent to the size of the JPEG x the number of frames in the animation.

(Quality does not apply to AVI or GIF files.)

10.3.4. MAX -> MIN or MAX -> MIN -> MAX

MAX->MIN or MAX->MIN->MAX

Most video players are able to generate a 360 ^{0} modeshape animation from 180 ^{0} worth of frames, (the **MAX->MIN** case). For those which are not use **MAX->MIN->MAX** which duplicates the frames to produce a full 360 ^{0} worth.

10.4. LASER PLOTTING

10.4.1. Introduction to Laser Plotting

Introduction to Laser Plotting

By default all graphics images generated by D3PLOT are sent only to the screen, but you can choose to copy them to laser files (postscript and pdf files for a laser printer).

This is done by pressing the "Plot" button when you are in Postscripts/PDF.

Laser language and file format used

At present D3PLOT writes Postscript laser files, using PS ADOBE level 2.0 commands, and PDF files. These are ASCII files that can be viewed and edited using any common editor.

"Encapsulated" Postscript files are not written, but later in Creating Encapsulated Postscript (EPS) files the very simple edits required to convert a file to encapsulated form are given.

Laser output is switchable between A4 (297 x 210mm), A3(420 x 297mm) and US "letter" (11" x 8.5") paper sizes. However the Postscript language makes it easy to edit files to fit other sizes.

The laser driver defaults to "PDF" file format, you can opt for "Postscript" laser file.

Number and orientation of plots on a page

The laser driver defaults to "landscape" orientation, with one plot per page. You can opt for "portrait" orientation and, in both cases, put multiple plots on a page in a variety of layouts.

Resolution setting

Postscript and pdf files generated can have higher than current screen resolution. Screen resolution, twice and four times screen resolutions are all available.

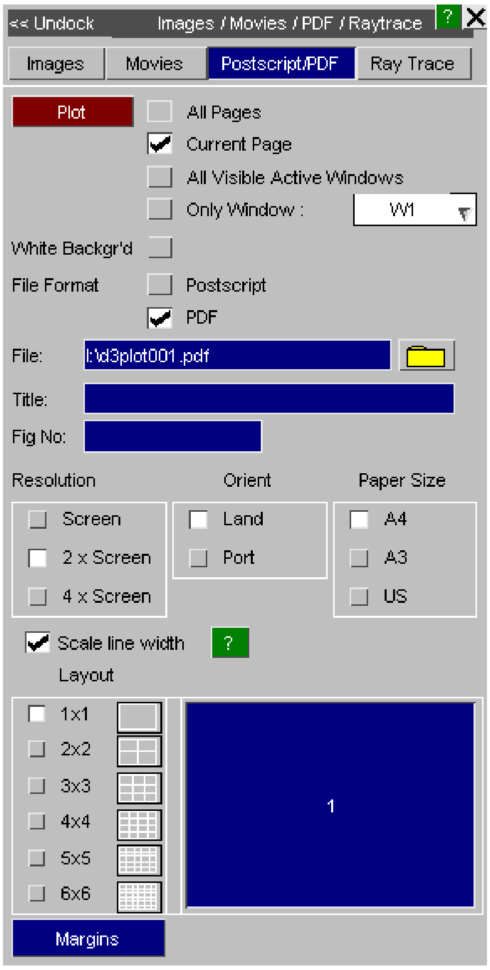
10.4.2. LASER Controlling Laser Plotting Using the Laser Plotting Panel

LASER Controlling Laser Plotting Using the Laser Plotting Panel

This figure shows the basic laser plotting panel.

This is invoked by the Postscript/pdf command under Images->Write in the top menu box.

It both controls and shows the status of the current laser file (if any).



Plot button

Press "Plot" button when you want to plot the current view on the screen into a postscript or pdf file.

Any plot directed to laser file is sent by default to the next free sub-image (if the file has multiple plots per page), or file (if only a single image per file, or the multiple page is full).

When multiple sub-images in a file are in use the next image to be written is shown by depressing the appropriate icon in the file layout panel. You can override this and choose a different sub-image: see Layout Controlling the number and layout of sub-images below.

Choosing the laser filename

When no file is currently in use the **File:** entry box will be available. You can give any valid filename for the next laser file to be written, or let D3PLOT choose one for you.

You can also use the  button to select a file via the standard file filter box.

If the file already exists you will be queried to check that you genuinely want to overwrite it: you cannot append to existing laser files.

The default naming convention used by D3PLOT for laser files is `postNNN.pdf` , where:

`NNN` is a 3 digit number (with leading zeros if required) in the range `001 - 999` .

Any existing files are skipped when the next file in the sequence is computed, so in the example above file `post001.ps` already exists.

Defining a label and figure number for laser plots

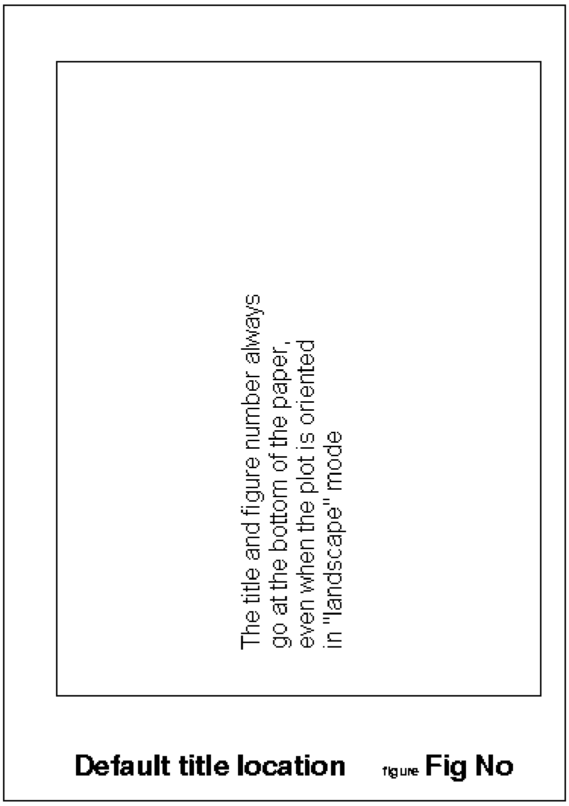
By default laser files are not labelled and have no figure number, but you may add either or both of these. They are always put at the bottom of the page, along the short edge, regardless of the orientation used for plots.

This figure shows the standard locations for title and figure number on laser plots.

The title may be up to 80 characters long, and is split over two lines if necessary by D3PLOT.

The figure number may be any string (not just a number), and is preceded by the word "figure". It is suggested that it is 6 characters or less long: here "12a" was used.

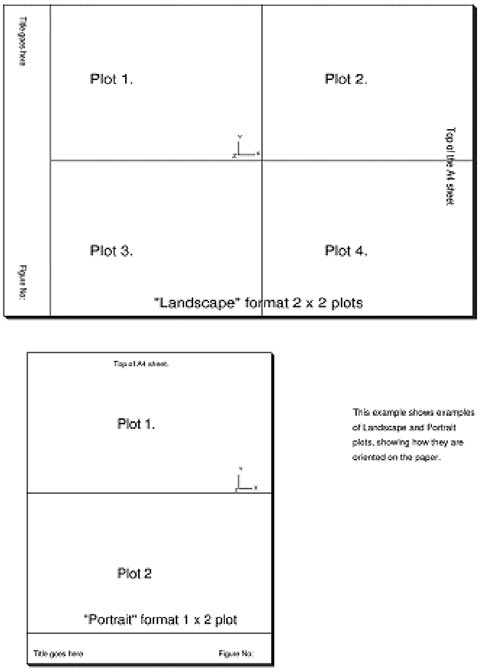
This plot is written in "landscape" format, and reinforces the point that the title and figure number always go at the bottom of the paper, regardless of the orientation of the plot contents.



Orientation Setting Landscape or Portrait plot orientation

By default plots are in "Landscape" orientation, with the long side of the plot aligned with the long side of the paper, but you can choose "Portrait" format instead.

The figure below shows examples of both landscape and portrait format plots, showing how they are aligned on the paper.

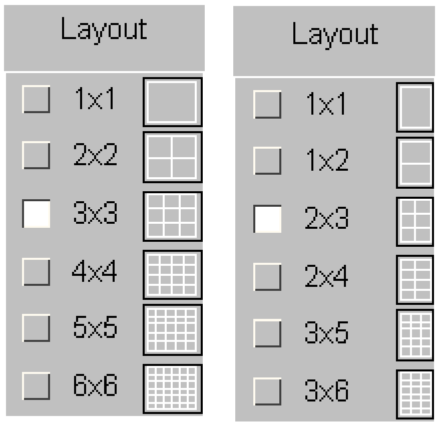


Layout Controlling the number and layout of sub-images

In both landscape and portrait formats it is possible to have more than one plot on a page.

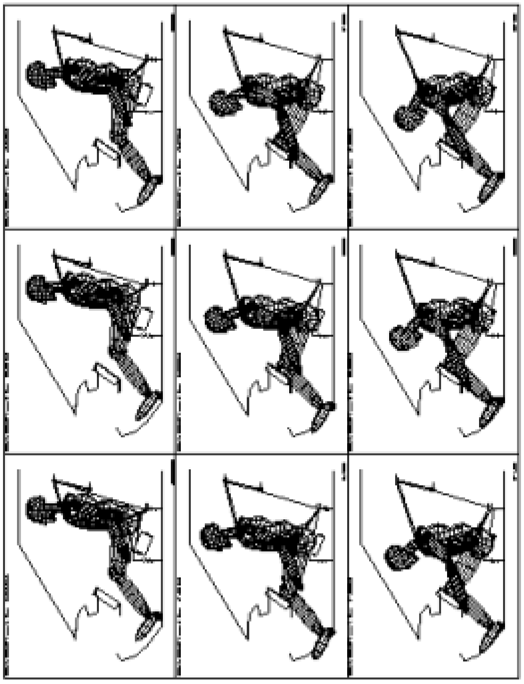
Various pre-programmed permutations of <#x> x <#y> plots are available as shown here.

Each individual plot on a page will be referred to from now as a "sub-image".

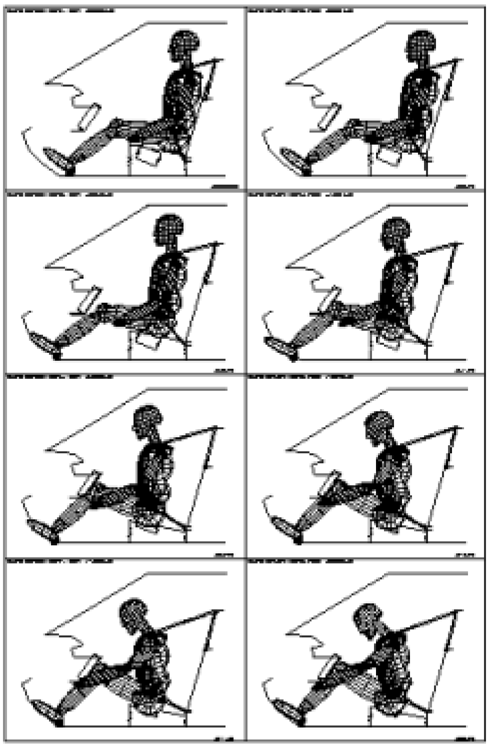


LANDSCAPE PORTRAIT

The figures below show examples of 3x3 Landscape and 2x4 Portrait multiple plots.

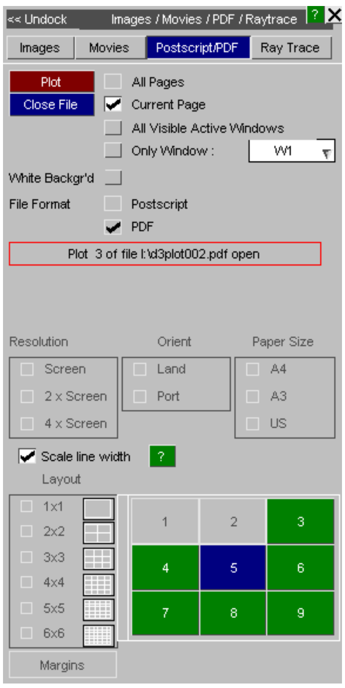


EXAMPLE OF 3 x 3 LANDSCAPE OUTPUT figure 7.1.5a



EXAMPLE OF 2 x 4 PORTRAIT OUTPUT figure 7.1.5b

Controlling the order in which multiple plots are drawn



The right hand menu shows a typical laser panel for a 3x3 portrait plot in which sub-images 1 and 2 are complete.

Normally sub-images are written in the order #1 to #n, but if the user wanted the next plot to be drawn to sub-image #5 instead of #3, they would click on the [5] icon where the button gets coloured in blue instead of the [3] icon as it normally would.

Next sub-image would be the next free one, i.e. #3 to receive the next plot. The [3] icon will be coloured in blue.

The status of files, and sub-images within files

D3PLOT laser files, and sub-images within files, have one of three possible states.

Inactive	**Green**	No graphics written yet, and not selected for the next plot.
Selected	**Blue**	No graphics written yet, but selected to receive the next plot.
Closed	Greyed out	File/sub-image complete, and cannot receive any more information.

The colours referred to above are used for the button icons on multiple sub-image panels, as shown in the figure above. Only green icons (ie those which are currently inactive) may be selected to receive the next image.

How sub-image status affects the destination of graphics

(1)

If no graphics have been written to a sub-image then the next plotting command will send laser output to the the sub-image currently "selected".

By default this will be the lowest numbered sub-image that has not yet been written to, but you can choose another as described above.

(2)

Once graphics have been sent to the sub-image its status changes to "closed" This means that it cannot receive further graphics.

Interaction between sub-images and files

A file with only a single image in it is treated in exactly the same way as an individual sub-image above, except that it is (implicitly) always "selected" for plotting until something is drawn in it.

A file with sub-images remains current (ie open) until all of the sub-images in it have been "closed", or the user closes it prematurely with a **CLOSE FILE** command. Then D3PLOT defaults to the next default filename as defined in Choosing the laser filename above.

The importance of closing files

While a file is still current it is still connected to the programme, and at least some of its contents will still be held in system buffers. If you want to send it to a printer you must close it first using a **CLOSE FILE** command.

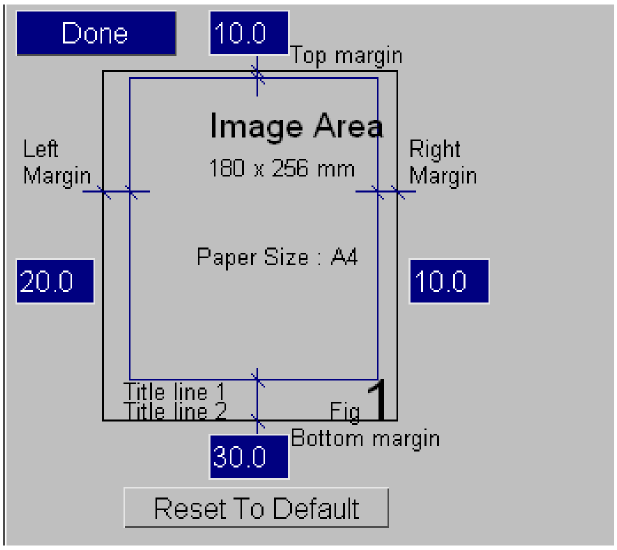
This flushes any remaining data to disk and disconnects the file from the programme.

10.4.3. MARGINS... Modifying Laser Paper Size on the Page

MARGINS... Modifying laser paper size on the page

The **MARGINS** button in the laser control panel gives a special sub-menu that allows you to select the margins on all sides:

The margins will only apply to the axis of the plot that comes closest to the paper borders; the other axis margins will be overridden to maintain the correct aspect ratio of plots (ie no image distortion).



10.4.4. Laser Plotting During Animation

Laser plotting during Animation

It is possible to store the sequence of images generated for animation in laser files. (This was not possible in earlier releases.) The figures showing [3x3 Landscape and 2x4 Portrait multiple plots]are examples of laser files created in this way.

To do this proceed as follows:

- Turn the laser switch on in the static part of the programme.

- Use **-> ANIMATE** to enter animation. You are warned that laser output is on, and have to confirm that it is genuinely required.
- Generate the animation sequence in the normal way. As each frame is created it is copied to laser the file just like an ordinary (static) plot.

The header on each frame is altered to show the animation frame number and corresponding time value. (Normally this would show the analysis title.)

The plot title is unchanged, but the figure number is replaced with the first frame number on each sheet.

Filenames are generated automatically if required using the standard naming sequence described in Choosing the laser filename .

10.4.5. Creating Encapsulated Postscript (EPS) Files

Creating Encapsulated Postscript (EPS) files

EPS format is used by many software packages to import postscript images. The laser files written by D3PLOT are not in EPS format, but only two very simple edits at the top of the file are required to change this.

The first seven lines of any D3PLOT laser file look like this:	To convert it to EPS format you must add a " **%%BoundingBox:** " line, and delete the " **statusdict** " line. Thus this file becomes:
%!PS-Adobe-2.0	**%!PS-Adobe-2.0**
%%EndComments	**%%BoundingBox: 0 0 595 842**
%%Pages: 1	**%%EndComments**
%%Page: 1 1	**%%Pages: 1**
	%%Page: 1 1
statusdict begin	**/a1test save def**
/a1test save def	

The arguments of the "BoundingBox" line are the Postscript coordinates:

<lower left> <lower right> <upper left> <upper right>

These must be expressed in raw Postscript space of 72 points per inch, and they assume that the paper is in portrait format with its origin at its lower left corner.

The the values in the example above refer to A4 format: 210 x 297 mm = 595 x 842 points; US "letter" paper would give 8.5" x 11" = 612 x 792 points. Clearly a smaller bounding box would select only a subset of the image.

For more information on encapsulated postscript see the "PostScript Language Reference Manual, 2nd edition" by Adobe Systems Incorporated. (Published by Addison Wesley, ISBN 0-201-18127-4)

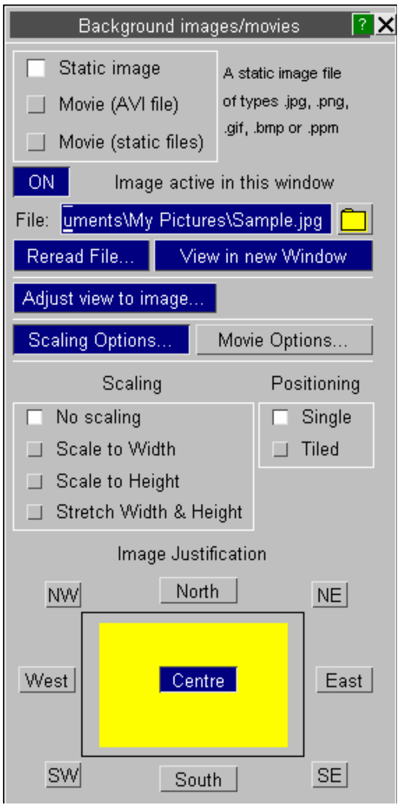
10.4.6. Notes on Laser Plotting

Notes on laser plotting

- Users on 3D devices should note that turning the laser on will temporarily force the graphics mode back to 2D. This is because a laser plot is intrinsically a 2D image and is computed in software.
- Transient graphics added "dynamically" to the screen are never copied to laser files. Examples are cursor-pick symbols, and also the information added interactively with the **DYNAMIC_LABEL** function.
- If an attempt to open a laser file fails because the file/directory refuses "write" permission, or the disk is full, you are warned and laser output is switched off.
- You can switch laser output **off** and **on** at will in the course of assembling a file with multiple images. Sub-images will only be written when the laser is on.
- Some of the defaults here may be preset outside D3PLOT via preferences in the `.oa_pref` file: see Appendix B

10.5. Reading Static Images and Movies

Reading static images and movies



Static image

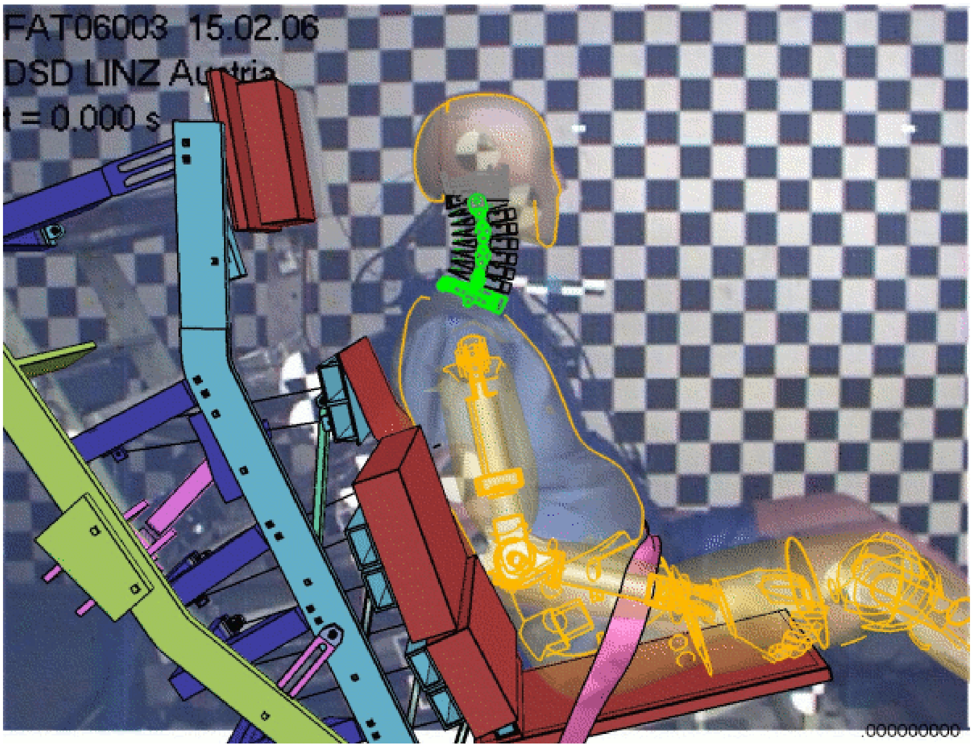
Read an image file to display as a background image behind a model instead of a solid background colour.

The formats we support are the same as we are able to write, see Capture .

Scaling Options

If the image dimensions do not match the graph window dimensions then the image can be scaled to fit or it can be tiled.

Below is an example background with the model overlayed on top.

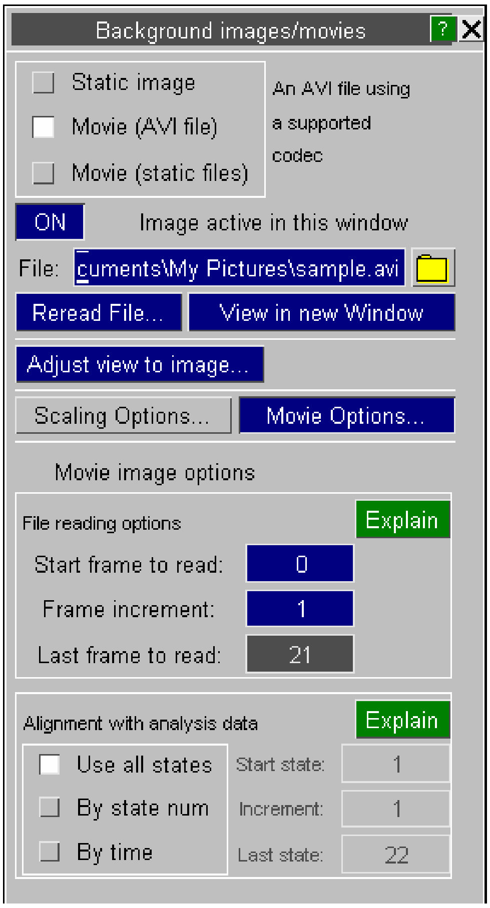


Movie (video file)

Read movies in one of the following formats:

MP4 (.mp4)

AVI (.avi)



We support all movie formats that we are able to create, namely MP4, bitmap AVI, and MJPEG AVI. Some AVIs need extra codec to be read in successfully, an error message will tell you that the codec is not supported and you will need to install the appropriate codec in order to read the movie.

Movie Options

Movie Options allow you to choose the start frame and state as well as the interval for both the movie and the simulation analysis. This can be particularly useful when you try to synchronize the movie with the simulation analysis. See Matching for how this can be done.

Scaling Options

If the movie dimensions do not match the graph window dimensions then the image can be scaled to fit or it can be tiled.

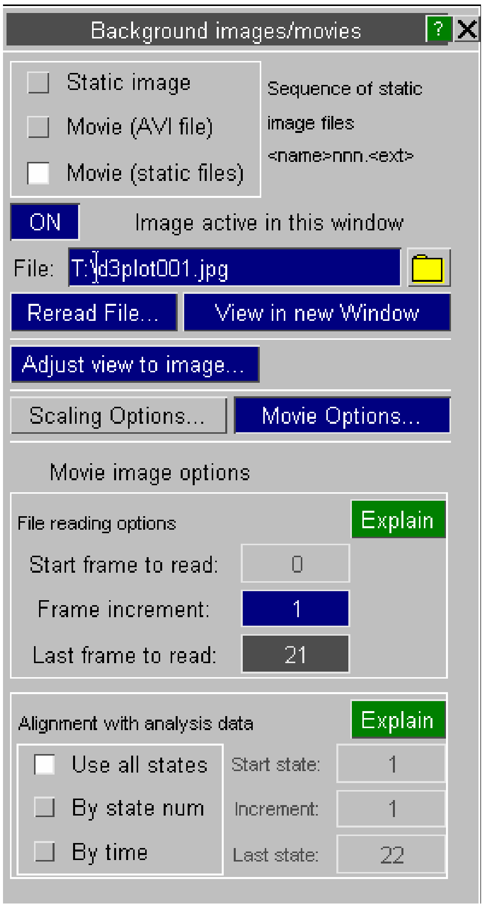
Movie (static files)

Read a series of files with the same name and extension in a format of <name>nnn.<ext>

E.g.

d3plot001.jpg, d3plot002.jpg, d3plot003.jpg.....d3plot010.jpg

D3PLOT will search for all qualified images and read them together. D3PLOT will then display them in the order as they're numbered as you play the simulation.



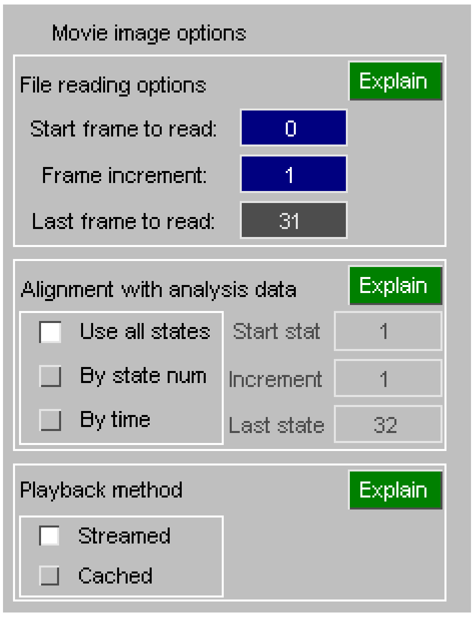
Movie Options

You can control the start frame and interval the same way as you do with movie frames, and aligh them with analysis data, too. See Matching for how this can be done.

Scaling Options

If the image dimensions do not match the graph window dimensions then the image can be scaled to fit or it can be tiled. This will be applied to all images once you've set it.

Further playback options



File reading options allow you to stipulate the start, increment and end frames to be read from the file. Each animation frame with have a single file frame displayed behind it, this allows you to choose what that should be.

Alignment with analysis data controls which frames from the analysis will be used. This is equivalent to using the main Anim > Set states popup to define the states to be displayed, and will modify the master animation display status accordingly.

Playback method controls whether background animation data is streamed from its source, or stored in memory (cached).

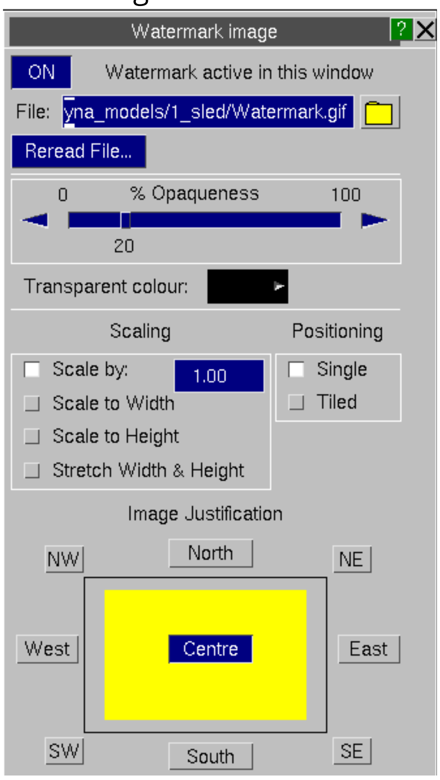
- **Streaming** decompresses each frame as and when it is required, so only requires storage for a single frame per window. This may be a little bit slower, although generally AVI files will decompress at a rate of 60 frames per second or better, but it require little memory.

- **Caching** decompresses all frames into an initial storage buffer, and then replays from this buffer. This is very fast, but if you have a significant number of frames in your background file you can end up running out of memory. This may be a suitable solution for a movie built from "static files", since they are likely to be few in number but relatively slow to decompress; however it is not recommended for MP4 or AVI files.

10.6. Watermarks

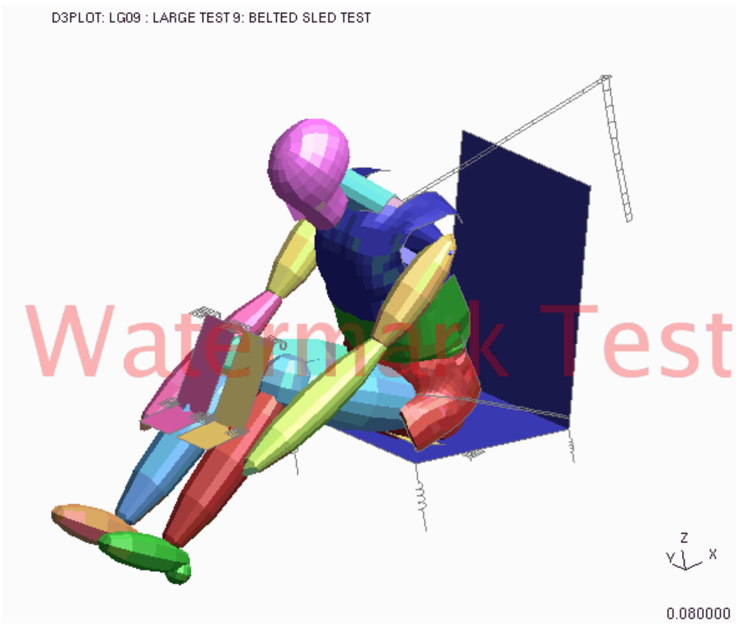
Watermarks

It is possible to add a "watermark" to a plot. Simply load in an image file in the watermark panel and set its transparent colour and overall transparency.



It will be drawn in front of the normal image, using the transparency settings you have defined. The position and size can also be set.

Below is an example with black as the transparent colour (the image was created on a black background) with 20% opaqueness.



10.7. Print... Option (Windows Platforms Only)

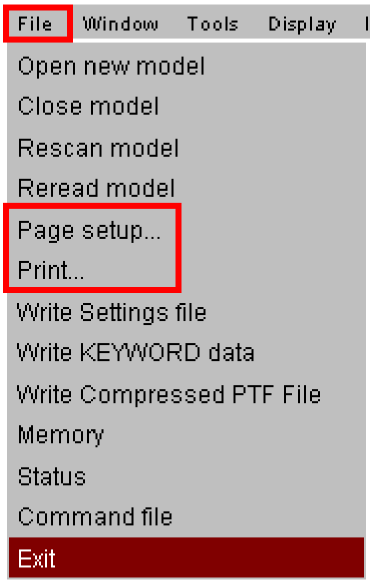
Print ... option (Windows platforms only)

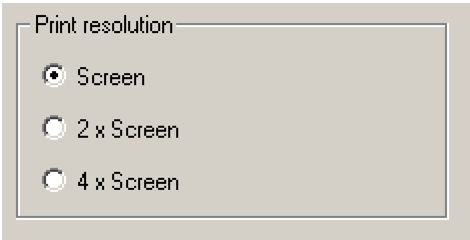
On Windows platforms only there are standard **Page setup** and **Print...** options under the **File** menu.

These will capture all currently visible graphics windows, size them to the current paper size rectangle and print them on the selected printer.

Print resolution:

On the standard **Print ...** panel there is an extra resolution option.





Choosing 2x or 4x resolution will capture the image at that factor times the current screen resolution, and will give a higher quality image. This may be useful if your screen is small, or you are planning to print on larger paper.

10.8. Ray Tracing

Ray Tracing

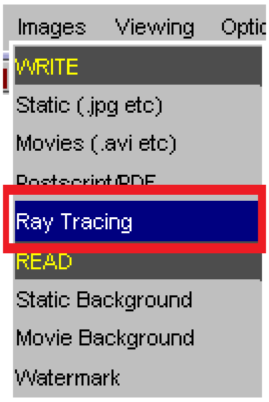
Rendering images using "realistic" lighting, including shadows, reflections and sophisticated visual properties.

This capability is new in D3PLOT 15.0, and very much in its infancy. It is *not*a "single button click to get a prettier picture" solution, and a fair amount of effort is required to get results that look impressive. It will continue to be developed, but at present it is quite crude.

It has been developed and tested on Windows, and while instructions for Linux download are given below this will only provide an older version that has not yet

been tested. Feel free to experiment on Linux, but it almost certainly will not work "out of the box".

[Go to usage explanation, skipping tutorial and installation instructions.]



10.8.1. What is Ray Tracing? A Mini Tutorial

What is ray-tracing? A mini tutorial.

Normal computer graphics, at least in the engineering world, use quite a simplistic approach to drawing objects: colour and transparency can be controlled, and also lighting, but the "surface properties" of the element being drawn are not usually controllable making it difficult to make much distinction between - say - steel, rubber, concrete and glass. In addition shadows are not cast, reflections are not seen, and light rays can pass through solid objects without hindrance. However it is fast, with modern graphics cards able to display and rotate huge models in real-time, making it ideal for interactive use on the desktop.

But there are times when you want to produce a higher quality image, and it becomes necessary to use a different approach.

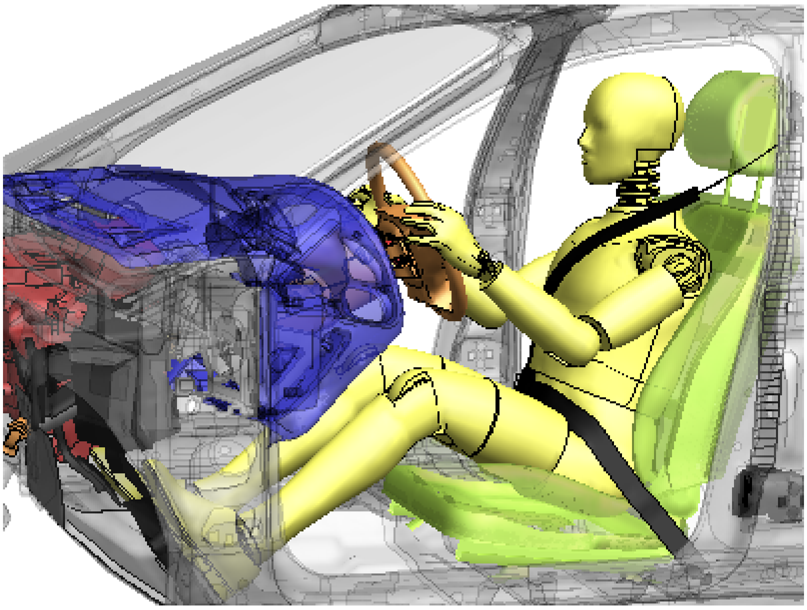
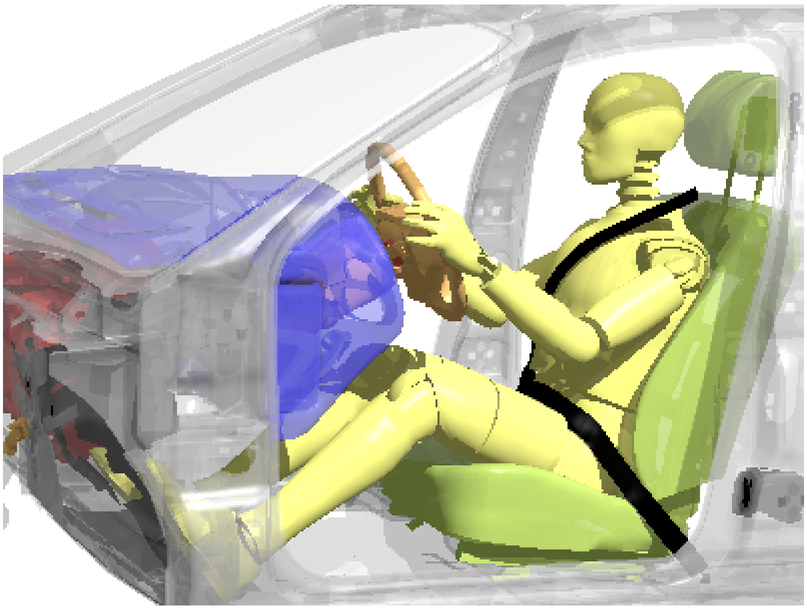


Image direct from D3PLOT
(Smooth shading used)



The same image rendered via ray-tracing
using default settings.

The examples above show raw D3PLOT output, and the same image rendered via ray-tracing. They are - deliberately - not very different, because the default settings attempt to emulate the D3PLOT visual properties, however you can see that in the right hand image there are shadows cast from the light source and a generally more realistic appearance. Also it corrects errors in transparency visible in the left hand image caused by deficiencies in conventional rendering (for example look at the lower seat cushion).

However there is no wireframe overlay, and other wireframe items, such as the belt upthe B post, are not drawn. In addition only SHaded mode plots are performed at present, so it is not possible to ray-trace contour plots.

So how is this done? As the name suggests ray-tracing considers "rays" of light bouncing off every part of the object, and it "traces"their path from object to eye, taking into account reflections, material surface properties and a host of other subtleties that are ignored in conventional computer graphics. An online search will reveal much more information, but a good starting point is [this wikipedia article].

10.8.2. The Implementation in D3PLOT Using POV-Ray

The implementation in D3PLOT: using POV-Ray

We do not perform these calculations inside D3PLOT, rather we export an intermediate file to an external Ray-Tracing package called "POV-Ray", which stands for the "Persistence Of Vision Ray tracer", and that performs the actual rendering in a separate external process.

POV-Ray is freeware that can be downloaded and used freely for any purpose, and indeed you will need to [download and install]it on your system in order to perform ray-tracing. The POV-Ray home page is [http://www.povray.org/]where you can find information about the tracer itself, its licensing, and also online documentation.

We have only just started to use this technology, and therefore our understanding of what is possible is quite limited. It is clear that absolutely stunning images can be produced, but when you start to use ray-tracing it also quickly becomes clear that creating these scenes requires a huge amount of effort, which is likely to be impractical for most of our users. Therefore our export of rendering data is quite simplistic, but the nature of POV-Ray input means that you can take this basic file and tweak it as much as you like, and we would welcome feedback about this process.

10.8.3. Installing POV-Ray

Installing POV-Ray

Download and install the POV-Ray package

This only needs to be done once, but to do it you will need the following:

- Access to a web browser and an internet connection.
- Permission to download and install software on your machine.

In a tightly controlled corporate environment it may be necessary to enlist IT support for this.

Go to http://www.povray.org/download/

On Windows:

- Download the "Windows Binary" via the " Download Windows Installer " link which (at the time of writing) will download version 3.7
- This is a conventional WIndows self-installer, so just save the file and then run it, following the instructions to install it on your system

On Linux: ***(note: we have not done any testing on Linux, it may not work...)***

- Go to the " old versions " page.
- Navigate to the Linux section and use the download link " Download (7.2 MB) via the web or via FTP " to obtain version 3.6 in binary form.
- Or you can download the 3.7 source code and build it if you are feeling brave...

Tell D3PLOT about where to find the executable.

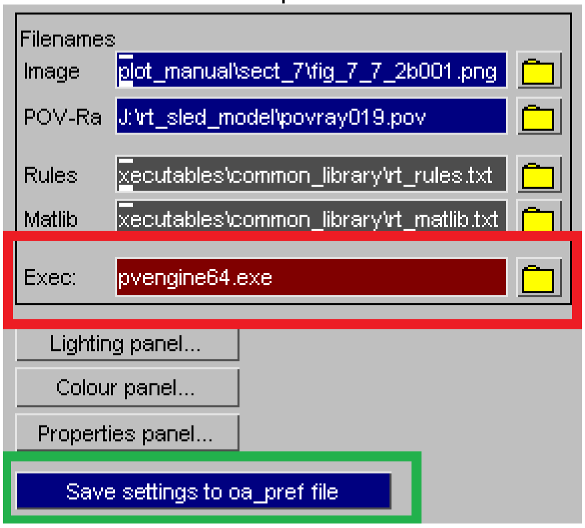
D3PLOT needs to know where to find the executable, which on Windows will be filename **pvengine64.exe.** By default it will be installed in **C:\Program Files\POV-Ray\v3.7\bin** , although you can choose a different location when you install the software. This needs to be stored as the preference

`d3plot*povray_executable:` *full_path_and_filename*

There are two ways of doing this:

(1) Interactively within the software itself:

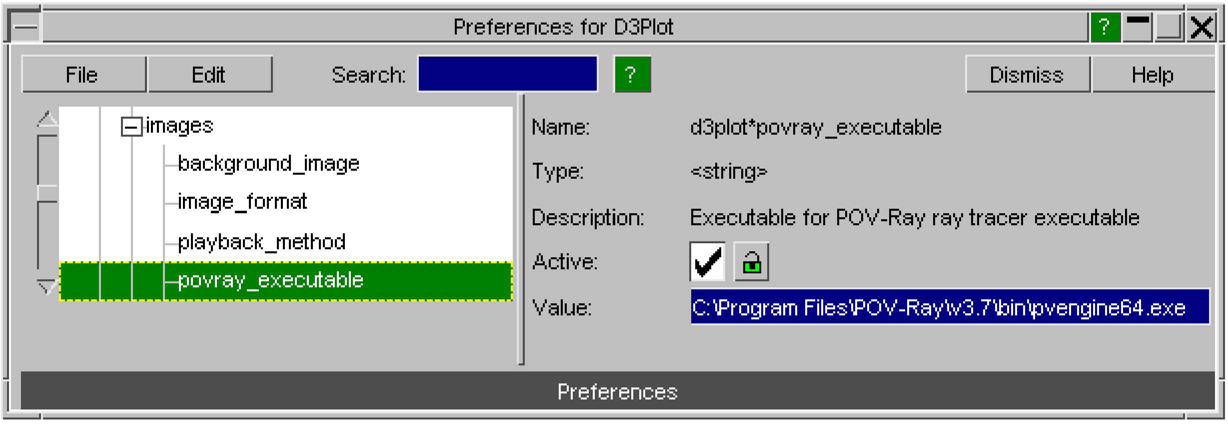
- Start D3PLOT and read in any model.
- **Images, Ray tracing** to map the ray tracing panel
- On the "Exec" line (outlined in red) use the file filter to locate the executable in its correct directory, which will update that line in white on grey.
- Use **Save settings to oa_pref file** (outlined in green) to update your home directory oa_pref file to contain this preference.



(2) Editing the oa_pref entry directly

- Using the **Options, Edit prefs** button bring up the preferences editor panel.

- Expand the **`[+]D3PLOT`** line
- Expand the **`[+]images`** line
- Click on **`povray_executable`**
- Tick the box to make it active
- Enter the full executable and pathname



10.8.4. Ray Tracing a D3PLOT Image

Ray-tracing a D3PLOT image.

The process gone through is:

- Select the window to be rendered. Only a single model in a single window is handled at present.

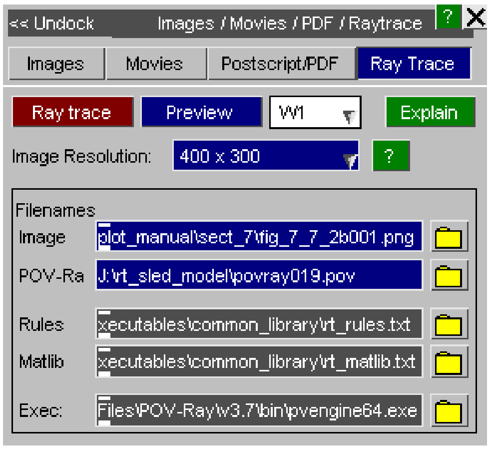
- Select the ultimate image resolution. This will default to the current D3PLOT window size, but other options are available including arbitrary sizes.

- Define the output image filename which at present is limited to type .png. A name will be provided automatically, but you can over-ride this.

- Define the intermediate POV-Ray input file, .pov. Again a filename will be provided automatically, but you can over-ride this.

- Define the "rules" file which is used to assign ray-tracing material properties to each part in the model.

- Define a "materials library" file containing material property definitions that you wish to use.



Finally use **Ray trace** to execute the process. This will apply the rules and launch the POV-Ray executable as a separate process to generate the image.

Defaults are provided for all the above so that no intervebtion is required to produce the default image. However once you start using this you will quickly become aware of the need to provide something other than the default ray-tracing material properties in order to obtain good results.

(The **Preview** button is an idea that has not come to fruition in D3PLOT 15.0, so it just prints a message stating this. The aim in future releases is to give a "fast but crude" preview of what the final image will look like using conventional D3PLOT rendering.)

Setting up the "visual universe": lights, camera position, etc

POV-Ray has the concept of a "universe" in which the model exists, defined as a coordinate system somewhere in space. The model to be rendered is specified in that coordinate system, and the camera (eye position, and where it is pointing) and lights are all defined in that system.

Fortunately D3PLOT has a more or less identical concept, and this is translated automatically for you into POV-Ray definitions. So as long as you are happy to replicate the D3PLOT view and lighting setup exactly you do not need to take any action as it "just happens".

However you may find that the differences between D3PLOT and POV-Ray lighting, and in particular the fact that in D3PLOT light rays can shine through opaque objects whereas in POV-Ray they cannot, can result in the "inside" of models looking rather dark. The crude solution to this is to increase the ambient light level, but too much of this "washes out" the scene, and a better solution may be to put a local light source inside the object to illuminate its contents.

Ray-tracing material properties

Within D3PLOT you have the following "material rendering properties"

D3PLOT simplistic material rendering properties	
Colour	A Red, Green, Blue (RGB) mixture, each 0 - 100%
Transparency	From 0% (opaque) to 100% (clear)
Diffuse Brightness	From 0% (dark) to 100% (full)
Specular Shininess	From 0% (dull) to 100% (very shiny)

You have no control over the "surface texture" of materials, which appear to be somewhere between dull plastic (low shininess) and metallic (high shininess).

However within a ray-tracer there is a much more sophisticated model. Colour and transparency are much the same, but surface properties are much more complex:

POV_Ray material rendering properties (or at least some of them)	

Colour and transparency	Defined by a " **`pigment`** ", typically
	
	`pigment { color rgb <r.r, g.g, b.b> }`
	`pigment { color rgbt <r.r, g.g, b,b. t.t> }`
	
	Where:
	
	• `r.r, g.g, b.b` are RGB colour values in the range 0.0 (dark) to 1.0 (full brightness)
	• `t.t` is transparency in the range 0.0 (clear) to 1.0 (opaque)

Brightness and shininess	Defined as components of a " **`finish`** ", typically `finish { ambient a.a diffuse d.d phong p.p }` Where: • `ambient a.a` is the material's response to ambient light in the range 0.0 (dark) to 1.0 (full brightness) • `diffuse d.d` is the material's response to direct light in the same range • `phong p.p` is the material's "shininess" in response to direct light, in the same range

	The "phong" value is approximately the same as D3PLO|T"s "shininess" value, but D3PLOT uses a simplistic model which combines both "specular brightness" (reflection of white light) and "shininess" (how concentrated those white highlights are). It is also possible to define "specular" values to control the concentration of white highlights.
Roughness	Part of a **`finish`** , typically: **`finish {`** *other attributes* **`roughness r.r }`** Roughness in POV-Ray approximates appearance on the scale 0.0 (shiny) to 1.0 (matt)
Reflection	Part of a **`finish`** , typically: **`finish {`** *other attributes* **`reflection { r.r }}`** Reflection ranges from 0.0 (no reflections) to 1.0 (reflects all incoming rays). Clearly high reflection means that the intrinsic colour of the item is not that significant, so high reflections should be used with low diffuse brightnesses.
Metallic	Part of a **`finish`** , typically: **`finish {`** *other attributes* **`metallic}`** This keyword uses an empirical algorithm to make surfaces look "more metallic".
... and there are plenty more attributes, but the above are the most common and - more importantly - the most easily understood.	

POV-Ray then assembles this **`pigment`** and **`finish`** into a " **`texture`** ", for example:

```
texture { pigment { color rgb < 0.2, 0.3, 0.4> } finish < ambient 0.2
diffuse 0.5 roughness 0.3 reflection 0.1 } }
```

and it this **`texture`** which determines how the object is rendered.

The POV-Ray #declare syntax, and associated conventions.

POV-Ray definitions can be quite verbose, so it has a #declare syntax to help with this, for example:

```
#declare P_Black_steel = color rgb <0.0, 0.0, 0.0>
```

Which creates simplified definitions of otherwise verbose definitions. For example after the `#declare` definition above you can use `P_Black_Steel` as shorthand for the full pigment definition.

The following naming conventions have been adopted by the POV-Ray community, and it is recommended that you comply with them:

#declare prefix	meaning
P_xxx	"pigment" definition
F_xxx	"finish" definition
T_xxx	"texture" definition

We have adopted these conventions in our standard material library file, see below.

So how do we create and assign textures to D3PLOT objects?

Assigning ray-tracing material properties using the "rules" and "matlib" files.

Ray-tracing properties are assigned on a per-part basis (it is not currently possible to assign properties to individual elements) and this is done using a pair of files

The "material library" file: rt_matlib.txt

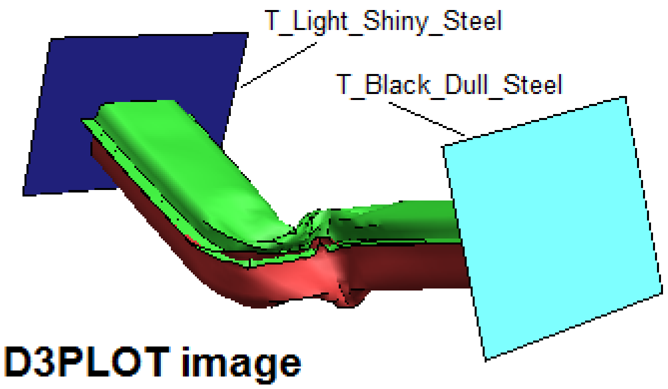
A default version of this file is provided in `$OA_INSTALL/common_library/rt_matlib.txt` , and D3PLOT is preconfigured to find it there. It contains two things:

- "#include" statements to include standard POV-Ray include files that are provided as part of the standard installation.
- Some very simple material definitions

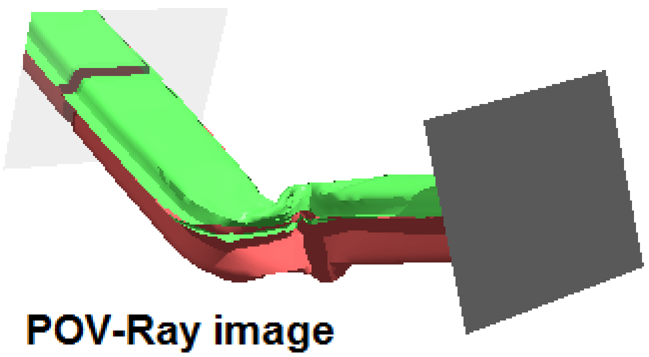
Examples of typical definitions within the file are

```
// Steel textures

#declare T_Black_Dull_Steel = texture { pigment { P_Black_Steel } finish
{ F_MetalA } }
#declare T_Black_Med_Steel = texture { pigment { P_Black_Steel } finish {
F_MetalC } }
#declare T_Black_Shiny_Steel = texture { pigment { P_Black_Steel } finish
{ F_MetalE } }

#declare T_Light_Dull_Steel = texture { pigment { P_Light_Steel } finish
{ F_MetalA } }
#declare T_Light_Med_Steel = texture { pigment { P_Light_Steel } finish {
F_MetalC } }
#declare T_Light_Shiny_Steel = texture { pigment { P_Light_Steel } finish
{ F_MetalE } }
```

Here is a simple example demonstrating how two of these textures might be used to replace intrinsic D3PLOT properties:



Original D3PLOT image, showing that the two end platens have received metallic properties for ray-tracing



Ray-traced image. The far platen is highly reflective polished steel reflecting the crush tube, the near platen is dull.

There are some other simple texture definitions in that file, and you could also explore the various include files (eg "metals.inc") that come with the POV-Ray install bundle. On a Windows machine these will be installed in

`%USERPROFILE%\Documents\POV-Ray\v3.7\include`

Where %USERPROFILE% is typically `C:\users\` *your_name* , although on a large corporate system with roaming profiles it may be on a network disk, for example `H:\users\` *your_name.* (To find the valueof environment variables on Windows open an MS-DOS (command) window and type "set".)

If you are at all ambitious about creating your own textures it is recommended that you copy that file to your own directory, add your own definitions and make this the current "rt_matlib" file for your own experimentation. You can do this by two methods:

1. Use the file selector to navigate to your private file and select it, making it current in the panel. Then use **Save settings to oa_pref file** to save this as a preference.

2. Set the preference `d3plot*povray_matlib_file:` *full_path_and_filename* directly.

The "rules" file: rt_rules.txt

This file gives a simple text-based method of assigning POV-Ray textures to parts in your model. A placeholder file containing its syntax and rules is provided in `$OA_INSTALL/common_library/rt_rules.txt` , and D3PLOT is pre-configured to look for it there, however that file does not contain any actual definitions, rather it is intended to be copied to a local area where you can set up your own rules. You can do this by two methods:

1. Use the file selector to navigate to your private file and select it, making it current in the panel. Then use **Save settings to oa_pref file** to save this as a preference.

2. Set the preference `d3plot*povray_rules_file:` *full_path_and_filename* directly.

Rules file syntax:

- Lines starting $, % or $ are treated as comments and are ignored

- Blank lines are ignored

- Data-bearing lines have the general form:

  `Keywordarguments= definition`

- Lines are processed in the order encountered, so if an earlier line is superseded by a later definition then that later definition "wins" and becomes current.

- Each line is free format, and keywords are not case-sensitive. However properties using #define should use the correct case.

- Where an entry is a single word it can be written verbatim, but if it contains white space it must be enclosed in "...". For example "rigid steel".

The purpose of this file is to assign POV-Ray textures to PARTs and MATerials in your model. Some lines define properties for PARTs directly, others refer to MATerials. In the case of MATerials these definitions propagate down to the constituent parts. You can think of it as working like this:

When a property is defined for a material:

```
For (each part of material)
{
Apply the property to this part
}
```

For anything other than Part names and labels in the table below to be parsed D3PLOT must have read a .ZTF file from PRIMER, since this contains the extra information (eg materials) that is not present in the normal d3plot / ptf file.

The following table describes the syntax in more detail. Entries in (...) are optional

Keyword	Syntax	Meaning
default	**default** = definition	All parts in the model get this definition. (See Rules for definition below) Example default = T_Black_Dull_Steel
part	**part** label_1 (to label_2) = definition	The part of label_1, or if label_2 is defined then all parts in the range label_1 to label_2, get this definition. Example part 1 = T_Glass_Clear part 2 to 20 = T_Light_Med_Steel

matl	**matl** label_1 (to label_2) = definition	The material label_1, or if label_2 is defined all materials label_1 to label_2 get this definition. This is propagated to all parts using these materials. Example matl 1 = T_Glass_Clear matl 2 to 20 = T_Light_Med_Steel
part_name	**part_name** *string_to_match* = definition	Assigns definition to all parts whose title matches *string_to_match* Example part_name rigid = T_Dark_Shiny_Steel part_name "moving & ductile" = T_Light_Shiny_Steel See the rules for *string_to_match* syntax below.
matl_name	**matl_name** *string_to_match* = definition	Assigns definition to all materials whose title matches *string_to_match.* This is propagated to all parts using these materials. Example matl_name rigid = T_Dark_Shiny_Steel matl_name "moving & ductile" = T_Light_Shiny_Steel See the rules for *string_to_match* syntax below.
matl_type	**matl_type** *ls_dyna *MAT name* = definition	Assigns definition to all materials using the LS-DYNA *MAT type name given. This is propagated to all parts using these materials. Example matl_type *MAT_RIGID = T_Dark_Shiny_Steel matl_name ELASTIC = T_Light_Shiny_Steel The LS-DYNA material name may use any of the following syntax:

		*MAT_ *name* eg *MAT_RIGID MAT_ *name* eg MAT_RIGID *name* eg RIGID The name matching is not case-sensitive, for example *MAT_RIGID and *mat_rigid will both work.
property	**property** type lo hi = definition Where type is one of **DENSITY** Material density **YM** Young's modulus **NU** Poisson's ratio **YIELD** Yield stress **FSTRN** Failure strain	Assigns definition to all materials for which property type has the valuelo<=value<= hi. This is propagated to all parts using these materials. Example property density 7.5e3 8.0e3 = T_Dark_Shiny_Steel property nu 0.45 0.5 = T_Rubber The lo .. hi values must be given in the correct order, and (obviously!) they must be in the right units. For example density in the example above is correct for steel in SI units of kg/m3.

Rules for string_to_match

String matching is not case-sensitive, and the test is "does this string match all or part of the part/material name?" A string_to_match may take any of the following forms:

A single word	eg steel
Multiple words separated by white space, which must be in "...", forming a string.. These must match exactly, including the number of white spaces. For example string "rigid steel" is not the same as "rigid steel".	eg "rigid steel"
Multiple words or strings separated by the logical operators:	eg rigid & steel

<table>
<tr>
<td>
<table>
<tr><td>&</td><td>Logical "and"</td></tr>
<tr><td>|</td><td>Logical "or"</td></tr>
<tr><td>!</td><td>Logical "not"</td></tr>
</table>

For example

<table>
<tr><td>rigid & steel</td><td>is true only if a name contains both words "rigid" and "steel"</td></tr>
<tr><td>rigid | steel</td><td>is true if a name contains either (or both) of "rigid" and "steel"</td></tr>
<tr><td>rigid & !steel</td><td>is true only if a name contains "rigid" but does not also contain "steel"</td></tr>
</table>

Expressions may also be put in (...) for example

(rigid & moving) | (elastic & stationary) will be true if a name contains both "rigid" and "moving", or if it contains "elastic" and "stationary". Brackets may be nested up to 20 deep, although if you are writing expressions that complicated you need to have a re-think!
</td>
<td>(rigid & moving) | (elastic & stationary)</td>
</tr>
</table>

Rules for definition

Recall that POV-Ray definitions are `textures` , that comprise a `pigment` plus a `finish` . The "definition" in the table above must be one of the following:

A texture This will be used verbatim, and all the internal properties in D3PLOT will be ignored.	For example: T_Clear_Glass texture { pigment {color rgb <0.7, 0.55, 0.3> } finish {ambient 0.2 diffuse 0.6 phong 0.4 } }
A pigment This will replace the colour used in D3PLOT, but the current transparency, diffuse brightness and shininess will be merged with it to produce a texture that is a combination of externally supplied pigment colour and internal D3PLOT properties.	For example: P_Darkish_Brown pigment { color rgb < 0.3, 0.4, 0.5 > }

A finish The colour and transparency in D3PLOT will be used verbatim, but the lighting properties will be replaced by this externally defined finish.	For example: F_MetalA finish { ambient diffuse 0.4 roughness 0.2 reflection 0.1 }

Suggested initial procedure

There is a lot to take in here, especially if you have no prior experience of computer graphics or ray-tracing. Experience suggests that the best way to proceed is as follows:

- Find a ***small*** model (ideally a few 100 elements at most), but one which has a reasonable variety of shapes with some curvature as it is hard to learn much from flat planes, and use that as a test-bed. The .pov files are verbose, and a small model will be much faster to work with.

- Copy the rt_rules.txt and rt_matlib.txt files from $OA_INSTALL/common_library to your local directory, and make these the current rules and material library files as seen by the ray-tracing panel. Then you can modify them at will without upsetting anyone else.

- Generate a .pov file using D3PLOT's default settings. This should produce an image, but what you actually want is the intermediate .pov file, as by editing that you can see almost instantly what the effects of changes will be.

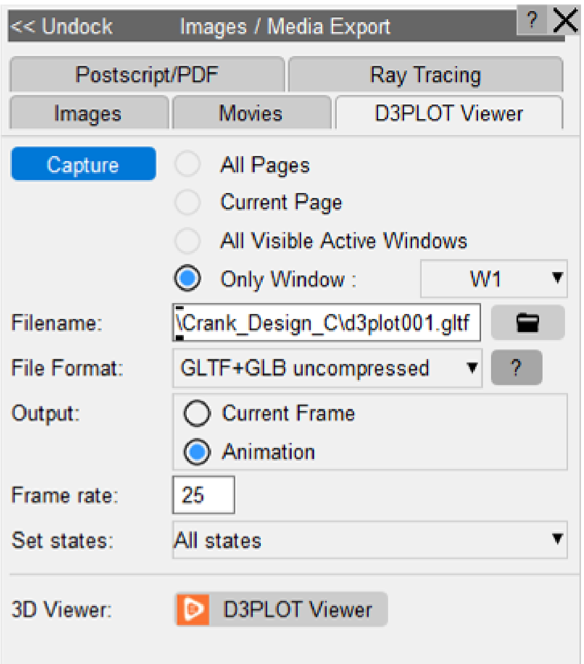
- Run POV-Ray manually and open this .pov file in its editor so that you can inspect its contents, then try editing and rendering the effects of your changes.

Once you get a feel for what changes you want to make try adding your modified textures to the rt_matlib.txt file, and then writing some simple rules in the rt_rules.txt file to apply them to your model, then use these to get D3PLOT to recreate a modified file for rendering. Only then try moving on to a full scale model.

10.9. D3PLOT Viewer Export

D3PLOT Viewer Export

From D3PLOT 18.0 onwards, you can export 3D files in GLB format for use in D3PLOT Viewer and other applications. D3PLOT Viewer export is available in a tab in the Images / Media Export panel.



10.9.1. Multiple Windows

Multiple Windows

Only one window can be exported at a time. If you have multiple windows, you need to select the window that you want to export using the drop-down menu.

Currently there is a limitation that only the first model in a window will be exported. Other models in a window can be exported by creating a new window containing only the model of interest, or by switching off all but the model of interest in **Window** >> **Edit window** >> **Wn** .

10.9.2. Compression

Compression

3D Viewer Export may be captured in one of two formats using the **File Format** drop-down menu:

GLB This is the default option: a GLB format file with Draco compression applied. This will compress data associated with solid, thick shell and shell elements (but does not compress the data associated with beam elements). It is slower to write and to read but the filesize is likely to be much smaller (5-10 times). **Microsoft PowerPoint does not support this compressed format.** GLB uncompressed This is an uncompressed GLB format file. Because no compression is applied, this format will be quicker to write and to read but the filesize is likely to be much greater (5-10 times). **Microsoft PowerPoint supports this uncompressed format in PowerPoint for Microsoft 365 or in PowerPoint 2019 or later.**

10.9.3. Animation

Animation

D3PLOT Viewer export can be captured either for a single frame or for an animation using the **Output** radio buttons.

3D animated output will only play in the D3PLOT Viewer. This is because the animation data is structured in a non-standard way to enable animated contour plots to be captured. Animated output will generally load into a standard GLB viewer and show only the first frame.

Use the **Frame rate** option to specify the default playback speed of an animated 3D export. You can also specify which states will be included in an animated export using the **Set states** drop-down menu.

10.9.4. Beam Elements

Beam Elements

D3PLOT Viewer export can capture beam elements either as lines or as true beam sections depending on the setting in **Display Options** . It is not possible to switch between the line and true beam section representations in the D3PLOT Viewer, so beams must be exported in the desired display mode. Also, note that a beam element captured with true beam section switched on will require more data to be written to the GLB file, which can result in significantly larger files.

10.9.5. D3PLOT Viewer

D3PLOT Viewer

D3PLOT 3D output is best viewed in D3PLOT Viewer, a web viewer that can be found at www.d3plotviewer.com and launched by pressing the **D3PLOT Viewer** button. Here you will be able to access functionality including:

- Navigating the model
- Playing animated output
- Blanking parts and includes
- Viewing the Part Tree
- Adding basic cutting planes
- Showing the contour bar

The files will load into other applications for example PowerPoint for Microsoft 365 and PowerPoint 2019 or later but some features will not be supported (including animation, contour bar and Part Tree). Please note that by default, GLB files are exported with Draco compression turned on. PowerPoint does not support Draco compression, so to show GLB models in PowerPoint, export as GLB uncompressed in the **File Format** drop down menu.

10.9.6. Limitations

Limitations

D3PLOT Viewer export has the following limitations:

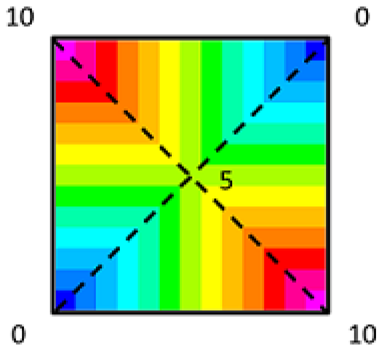
- Solid, thick shell, shell and beam elements are exported; other element types are not
- SH, CT and SI plot modes can be captured; other plotting modes revert to SH plot
- Contour values are only exported as a single value at each node, so only low and medium resolution contour modes are supported. GLB files written with contour mode set to unaveraged or high resolution will be captured at medium resolution
- Exporting cut sections is not currently supported and cut-sections are automatically disabled prior to exporting 3D output and then enabled again on completion

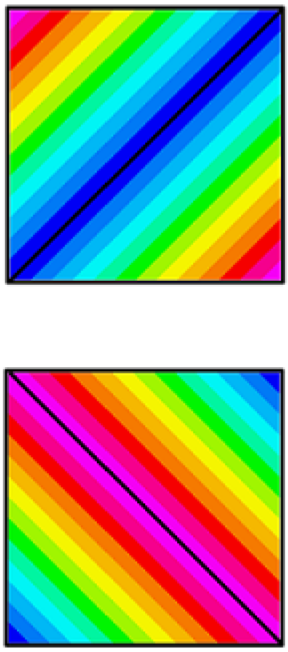
Additionally, there are limits to how much data can be read in by D3PLOT Viewer due to browser limitations. Uncompressed GLB files are limited to 2GB of data and any file larger than this is not able to be read by D3PLOT Viewer. Standard (compressed) GLB files can contain around 4GB of uncompressed data as their file size will be considerably smaller, but note that performance of D3PLOT Viewer will be impacted by large amounts of uncompressed data.

Contouring Faces with Contraflexure

When contouring quad faces with "contraflexure", where nodes on one pair of diagonally opposite corners both have higher values than the other two nodes, D3PLOT creates a point at the element centre with a value equal to the average of the node values and then subdivides the quad face into four triangles for drawing (see diagram below).

When a contoured model is exported to GLB format, quad faces are simply split into two triangles, which are then contoured based on the node values only, i.e. the element centre value is not included (see diagram below). This simpler approach reduces the complexity of the GLB file and also reduces the file size. When viewing contour plots in a large model, the differences are unlikely to be noticeable, however when viewed across a small number of elements, differences in contour patterns may be visible, as illustrated for a single quad face in the diagram below. The GLB representation is not technically incorrect, it is simply a different approach. Importantly, it does preserve the peak values, so, although the detail of the contour pattern may differ slightly from D3PLOT, the peak values will remain the same.





Contours in D3PLOT

Possible contours in GLB

11. Reloading Program States

Reloading Program States

It is often useful to save certain program settings and then be able to reload them at a later time to restore the program to a previous state, such as:

- the models loaded in D3PLOT
- the number of graphics windows
- the camera position
- the blanking status of entities
- the current display options

D3PLOT has a few different methods for doing this, each one saving and reloading a different batch of settings. Each method is described in the manual, but they are all in different sections making it difficult to compare each one.

This section aims to summarise each method in a single location so that they can easily be compared and make it easier for the user to decide which one is most appropriate for their situation.

11.1. List of Methods

List of Methods

There are currently five methods for saving/reloading programme settings. Links to the sections describing each one and how to use them are listed here:

- Session File: Session File and UTILITIES, SESSION_FILE
- Settings File: UTILITIES, SETTINGS_FILE
- Properties File: Properties: Controlling Colour, Drawing Style, Transparency, Lighting Attributes and Overlay of Entities , Saved properties and The Properties (.prp) file stores Model-specific information
- Template File: Template File
- View Files: VIEW MANAGER... Storing and Retrieving "View" Information

In addition to the above methods, from v20 D3PLOT has an annotations feature that uses settings and properties files to save the configuration of a page, including the filenames of the models on the page, and allow that configuration to be reloaded for each of multiple annotations.

11.2. Table

Table

Below is a table showing the settings that are saved by each method/file type.

As can be seen the Session file is the most comprehensive method as it saves everything contained in both the Settings and Properties files. In addition it saves the selected states of an envelope plot, whereas a Settings file will select all the states when it is reloaded irrespective of what was originally selected. If you want to restore a previous session of D3PLOT, for example if you want to close D3PLOT down at the end of the day and then restart it the next day, then this is the option to use.

Settings files save and reload programme and window status information.

Properties files save and reload model-specific information.

Prior to the introduction of Session files (in D3PLOT 16.0) using the Settings and Properties files together was the best way to restore a previous session. Using a Session file is the preferred method now as it is more user friendly. However, if you only want to restore programme/window status information then using a Settings file is the way to go. If you only want to restore model-specific information then use a Properties file.

Template files only save a small subset of settings, but are useful for controlling which windows models are located in and their colours.

View files only save the camera position and are useful for replicating views when capturing images.

	Session	*Settings*	*Properites*	*Template*	*View*
Number of Graphic Windows and Location of Models	x	x		x	
Model Colours	x	x		x	
Camera Position	x	x	x		x
Model Appearance (Colour, transparency, display mode, etc.)	x		x		
Blanking Status	x		x		
Components and Contour Settings	x	x			
Envelope Plot Settings	x (selected states)	x (all states)			
Display Options	x	x			
Cut Sections	x	x			
Measure	x	x			
Deform Menu Settings (except Transform)	x	x			
Transform	x		x		
Volume Clip	x	x			
Traces	x	x			
Window Background and Text Colour	x	x			

11.3. Diagram

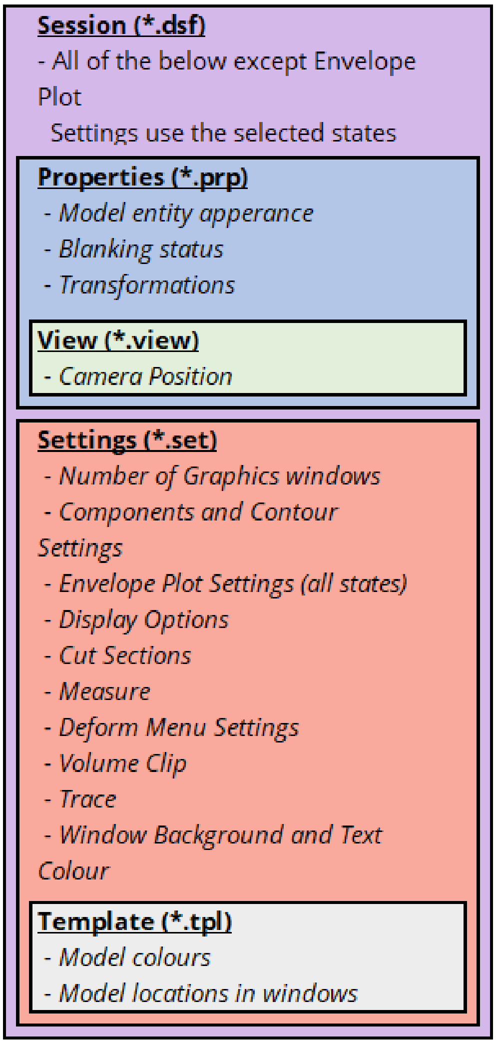
Diagram

Below is a diagram showing how each file type relates to each other.

View files are a subset of Properties files.

Template files are a subset of Settings files.

Properties and Settings files are subsets of Session files.



12. Display Options

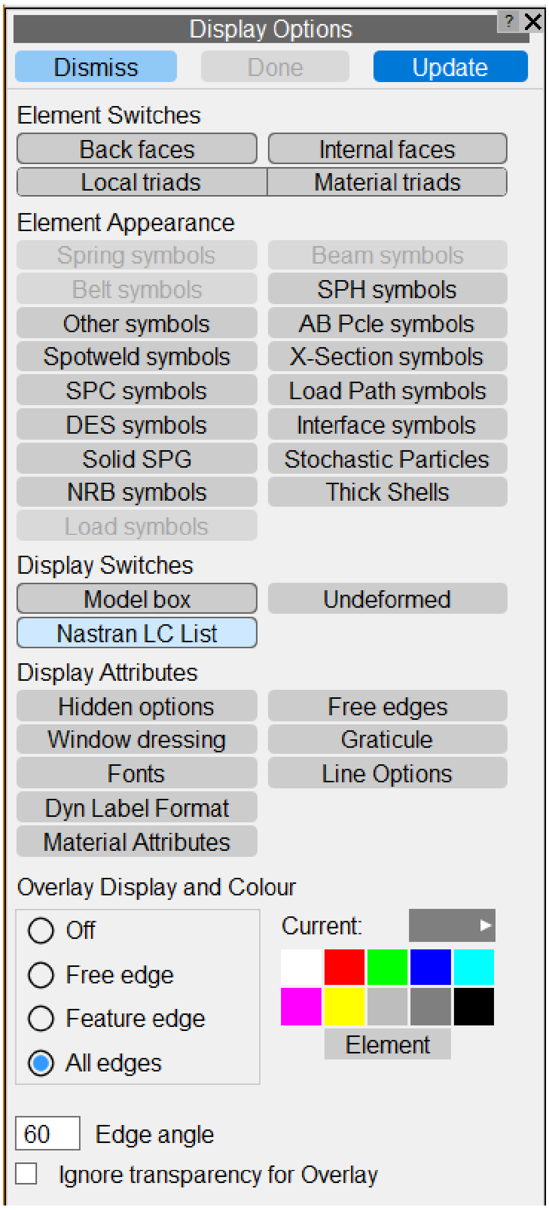
DISPLAY OPTIONS

Controlling the appearance of the display and many of the items drawn on it.

The main **Display Options** control panel is shown right.

This contains switches (left column), and options with sub-menus (right column), which control many aspects of display appearance.

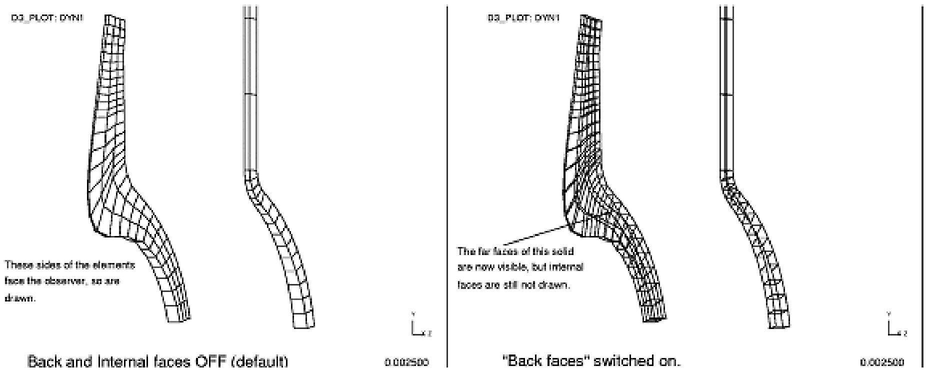
Overlay display, colour and edge angle and transparency are described in OVERLAY... Controlling the Hidden-Line Overlay of Element Borders On Data Plots



12.1. BACK_FACES Switch: Display of "Back" Faces of Solid and Thick Shell Elements

BACK_FACES switch: Display of back faces of solid and thick shell elements

To save graphics rendering effort the "back" faces of 3D elements (solids and thick shells) can be turned off. This is the default on 2D devices, but on 3D devices they are left on (in case the model is rotated to expose its back). The effect of this is shown in the figures below.



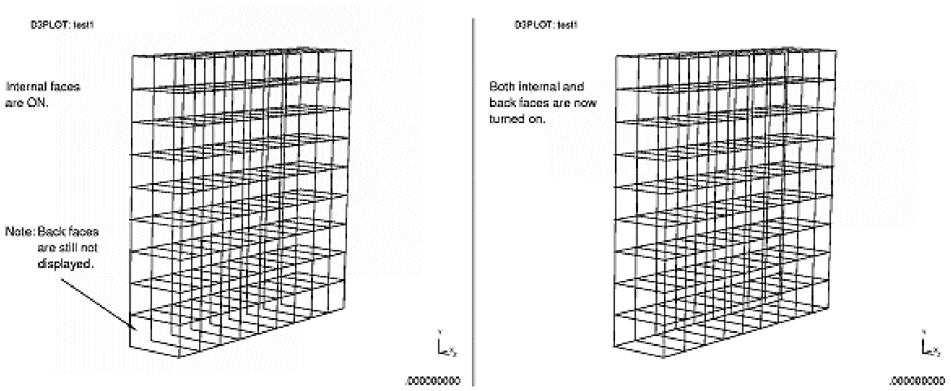
BACK_FACES off

BACK_FACES on

12.2. INTERNAL_FACES Switch: Display of Inside Faces of Solid & Thick Shell Elements

INTERNAL_FACES switch: Display of inside faces of solid & thick shell elements

By default the internal faces of 3D elements are never displayed: as well as requiring extra rendering effort leads to cluttered plots. The figure below left shows a wall of solids with internal (but not back) faces turned on, and below right, the same wall with both internal and back faces on.



You may need to turn internal faces on if you are trying to screen pick solids several layers deep by screen-area. Otherwise each screen area selection will only pick the (visible) solids at the front of the image, like peeling an onion, and several such picks will be required to select all the solids through the depth.

When D3PLOT is displaying a CT, SI or LC data plot then by default only the external faces of 3D elements are used to calculate the contour bar range. If internal faces are turned on then they are also included in this calculation. From version 18 onwards it is also possible to include the internal faces in the calculation without having to turn them on - see Visible faces vs All faces).

12.3. LOCAL_TRIADS Switch: Display of Element Local Axes

LOCAL_TRIADS switch: Display of element local axes

Solid, shell, thick-shell, beam and interface elements all have local coordinate systems, and you can display "local" triads of their axes using this switch. The figure below left shows examples of local triads on solids, and the figure below right, on beams and shells.

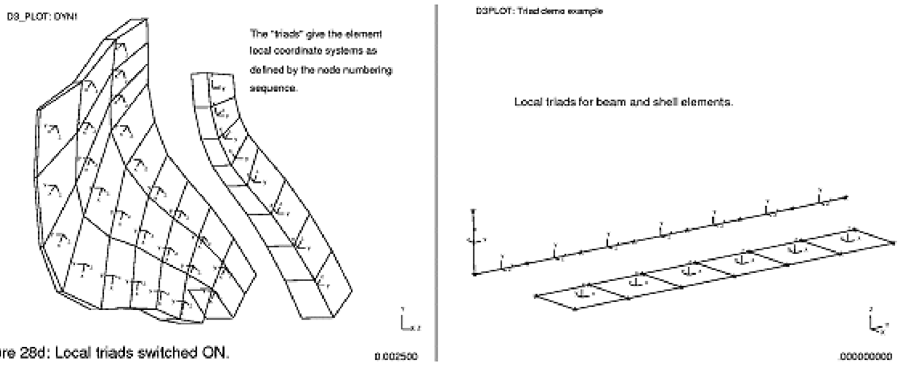


Figure 28d: Local triads switched ON.

The method used to compute element local axes is given in the following section:

<table>
<tr><td>Solids</td><td>Transforming Directional Solid Results to the Element Local System</td><td rowspan="4">In all cases D3PLOT generates local axes from the element topology alone.

Note that it does NOT currently "know" about any local material axes, beta angles or ply directions.

From D3PLOT 13.0 onwards a local ply X axis can also be plotted (see Ply local X axis).</td></tr>
<tr><td>Thin shells</td><td>Thin shell coordinate systems</td></tr>
<tr><td>Thick shells</td><td>Frame of Reference: Computing the Local Coordinate System</td></tr>
<tr><td>Beams</td><td>"Basic" Components for All Beams</td></tr>
</table>

Segments	Contact segments use the same method as thin shells.

An easier way of spotting reversed outward normals of shells and interface segments

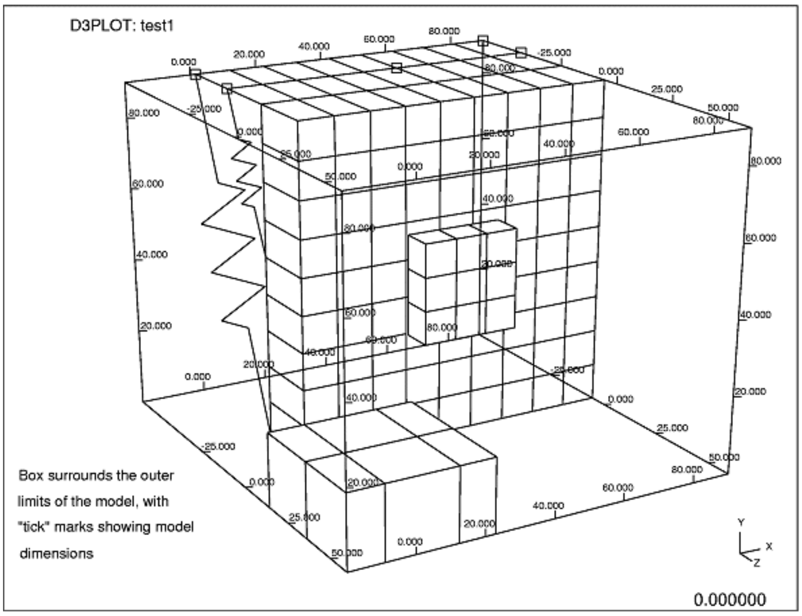
The local Z axes (outward normals) of shells and interface segments are important in some contexts. For shells averaging across adjacent elements with top and bottom surfaces flipped gives misleading contours, and for interface segments some types of contact surface cause problems if their outward normals face the wrong way.

Therefore, as well as being able to display local triads, you can plot their **OUTWARD_NORMAL** components in **CT** continuous-tone plots. This plots normals that face you in magenta, and those that face away from you on blue. It makes it easy to spot elements that face the wrong way: certainly it is easier than hunting for reversed local Z axes using triads!

12.4. MODEL_BOX Switch: Displaying the Model External Dimensions

MODEL_BOX switch: Displaying the model external dimensions

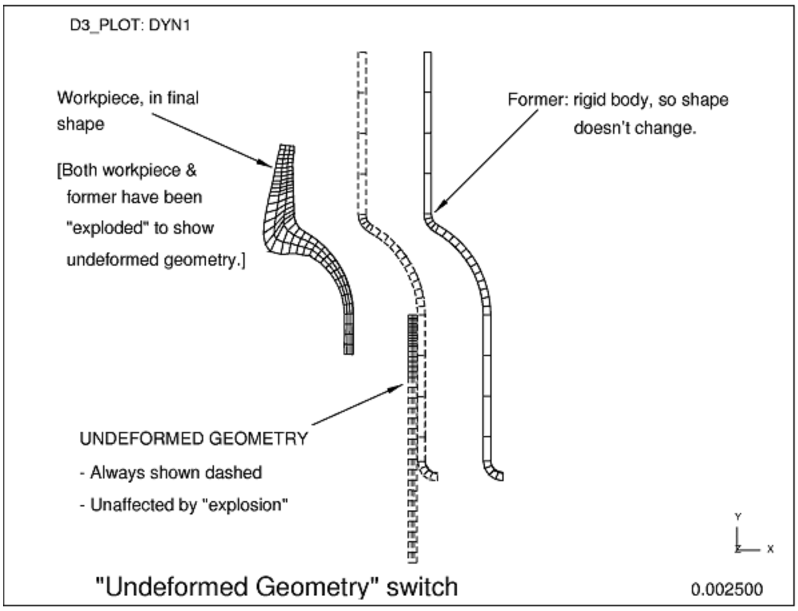
It can be useful to know what the external dimensions of your model are. One way to do this is to use a **GRATICULE** (see Graticule), but this uses screen space and can't be transformed with the model. The **MODEL_BOX** option is an alternative: it draws a box that represents exactly the largest [x,y,z] external dimensions of your model, and adds tick marks. The figure below shows an example of this. This box rotates with the model, and can be viewed from any angle.



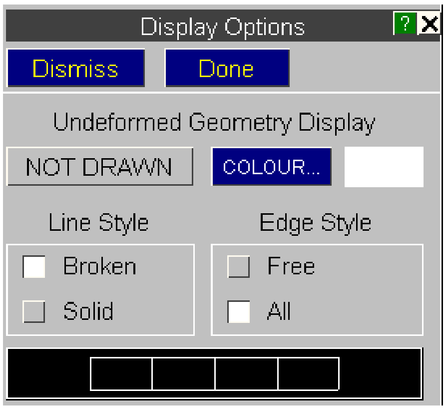
12.5. UNDEFORMED... Menu Displaying the Undeformed Geometry

UNDEFORMED... Menu Displaying the Undeformed Geometry

Normally just the current geometry is drawn, but you can overlay this with a hidden-line plot, using in various styles, of the Undeformed geometry. An example is shown below.



The following attributes of the undeformed geometry may be set:

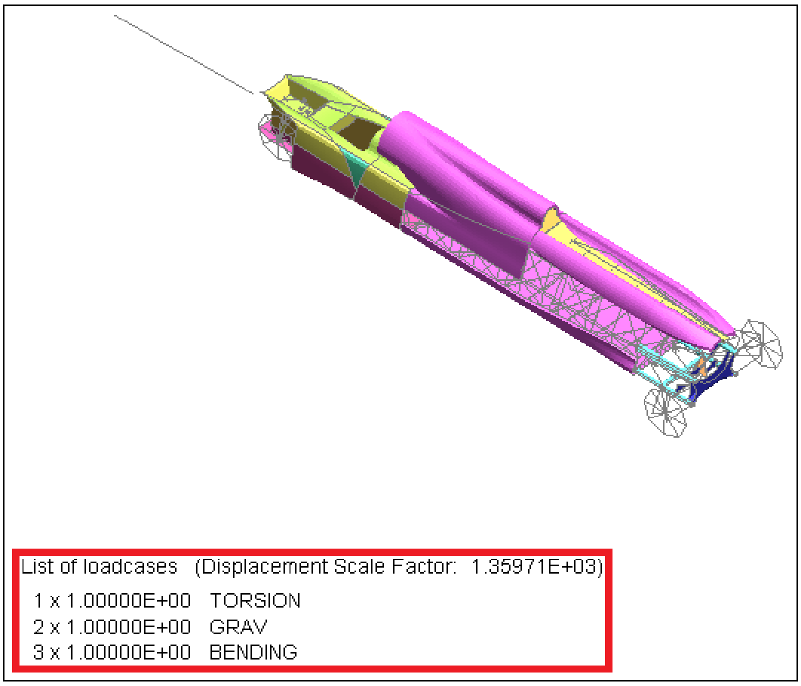


(NOT_)DRAWN	Whether it is displayed.
COLOUR	Its line colour
Line Style	May be **Broken** or **Solid**
Edge Style	May be **All** edges or **Free** edges only.

12.6. NASTRAN LC LIST

NASTRAN LC LIST

For Nastran models D3PLOT will normally write a list of loadcases in the bottom left hand corner of the graphics window. This can be turned off with this switch.

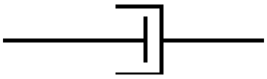


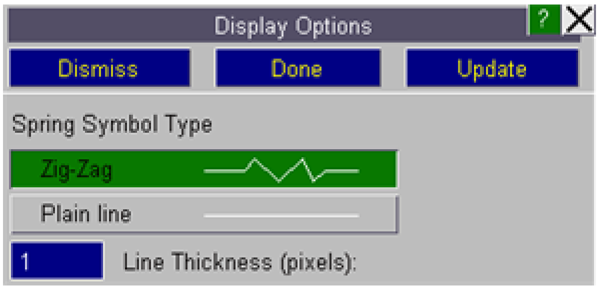
12.7. SPRING_SYMBOLS... Menu: Setting the Drawing Style for Springs and Dampers

SPRING_SYMBOLS... menu: Setting the drawing style for springs and dampers

By default springs use "zig-zags"

and damper elements use "dashpots"





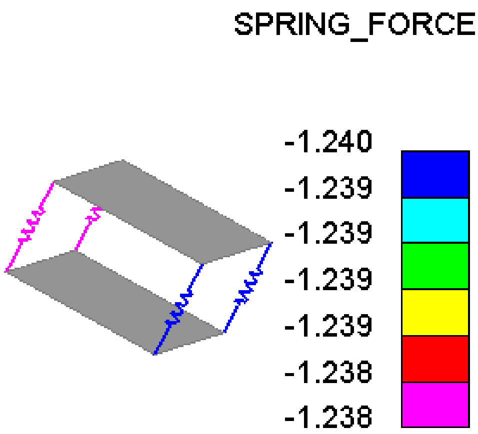
You can change these to **Plain line** style, in which they are drawn as simple lines. This saves vectors, reduces clutter, and can be useful during animation to improve speed and economise on memory usage. See SPH Symbols Managing SPH Element Display for details of how grounded and zero length elements are drawn

Line Thickness

By default springs will be drawn using the line thickness set in the Utilities Graphics menu. To highlight springs the default thickness can be set to a value in the range 1>10 pixels. This thickness will also be used when contouring spring forces.

If a model contains coincident springs with the same node numbering then the colour of the 1st spring that is drawn (the lowest ID) will be the one seen in the contour plot.

If a spring translational data component is selected then any rotational springs will be drawn uncontoured in grey. If the component is a rotational component then any translational springs will be drawn uncontoured in grey..

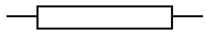


12.8. BEAM_SYMBOLS... Menu: Setting the Drawing Style for Beams

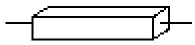
BEAM_SYMBOLS... menu: Setting the drawing style for beams

By default beams are drawn as lines

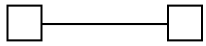
You can choose to use "thick" lines

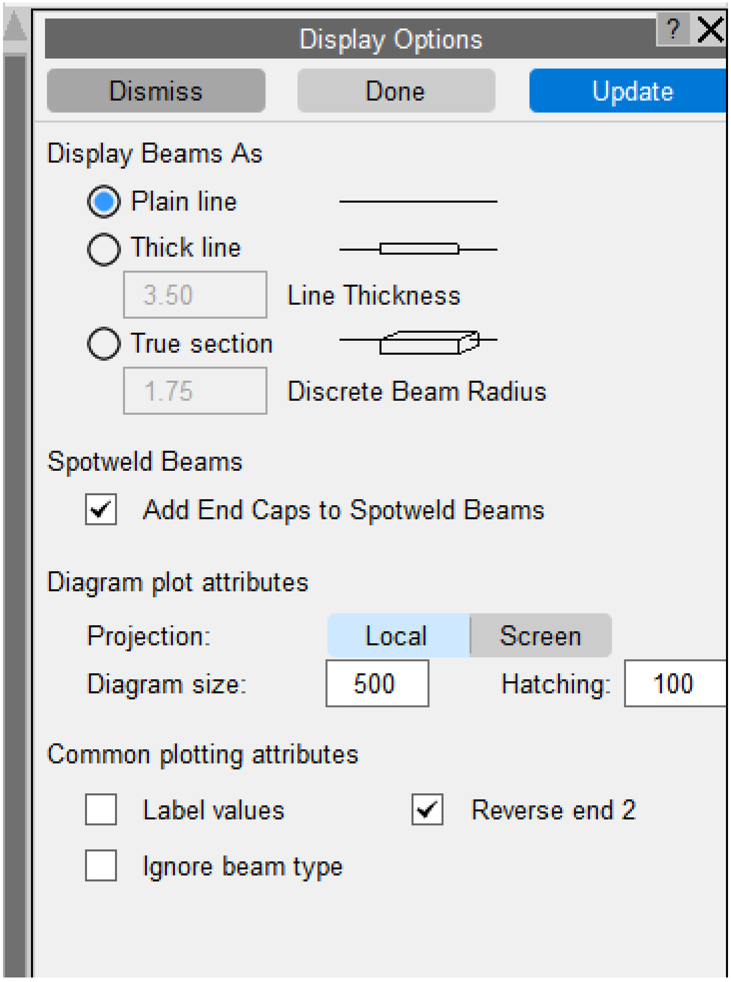


If a ZTF file had been read then beams can also be drawn using their true section



"Thick" and spotweld beam symbol sizes may be set individually.





12.8.1. Thick Line

Thick Line

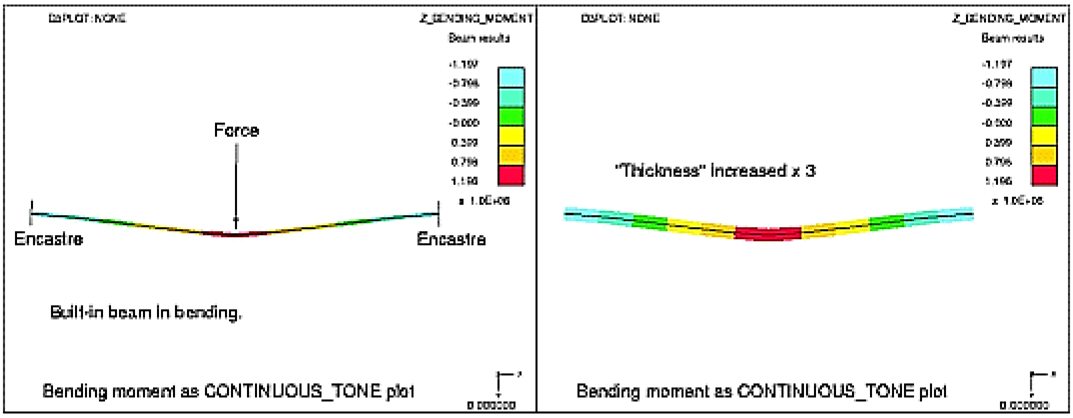
The "thick" symbols can make it easier to see beams in plots that contain a lot of lines, although they are slower to draw. The nominal dimensions in the boxes below are calculated as a %age of the overall model diagonal length, and they may require adjustment to give sensibly sized symbols for very asymmetrical models.

The "thick" line used will depend upon whether or not a .ZTF file has been read in from PRIMER.

- If it has been read then the beam's "true" section thickness will be used.
- If no such file has been read a square section of the "Thickness" given here will be used.

Continuous-tone beam plots show the data values on a beam as thick bars of solid colour along the beam centre-line. The thickness of these bars is set by default to 1% of the model's longest dimension (in model space units), which proves adequate for most models.

The figure below demonstrates the effect of changing this value by a factor of 3.



12.8.2. True Section: Displaying the Actual Beam Cross-Section

True Section: displaying the actual beam cross-section

If a .ZTF file from PRIMER 9.2 or later has been read, section information will be available for beams which, if this option is selected, allows them to be drawn using their "true" section shapes.

For "Integrated" beams with explicit section dimensions this is simple, and the actual dimensions are used.

For "Resultant" beams where only Area, Ixx and Iyy properties are available then a thin-walled rectangular section that matches these properties is synthesized. This should be approximately correct, but obviously it cannot represent I beams or rectangular sections with varying wall thicknesses, but it should give a reasonable representation of beam dimensions. If you use inconsistent or impossible properties you may get some strange looking sections!

If True Sections are being plotted D3PLOT can not calculate a size of shape for discrete beams. By default discrete beams are drawn as a sphere with a diameter that is 1% of the model's longest dimension (in model space units). This size can be adjusted if necessary.

True Section: Recommended modelling practices

There are a number of modelling choices that can be made when defining beam elements which can affect how they are displayed in D3PLOT. The following practices are recommended for the best visualisation of results in D3PLOT.

- Define the beam orientation using a unique 3rd node for each beam rather than the _ORIENT option.

- Set NREFUP=1 on the *CONTROL_OUTPUT keyword. This will update the position of each 3rd node through the analysis, allowing D3PLOT to display the correct rotation.
- If the _OFFSET option is used to define an offset, ensure that it is specified in the positive direction of the orientation vector. If it is specified in the opposite direction the beam offset can be shown as flipped in D3PLOT (the LS-DYNA analysis will still be correct).

12.8.3. Add Caps to Spotweld Beams

Add Caps to Spotweld Beams

Similarly the size of the "blobs" on the end of spotweld beams may need adjusting to give visually acceptable results.

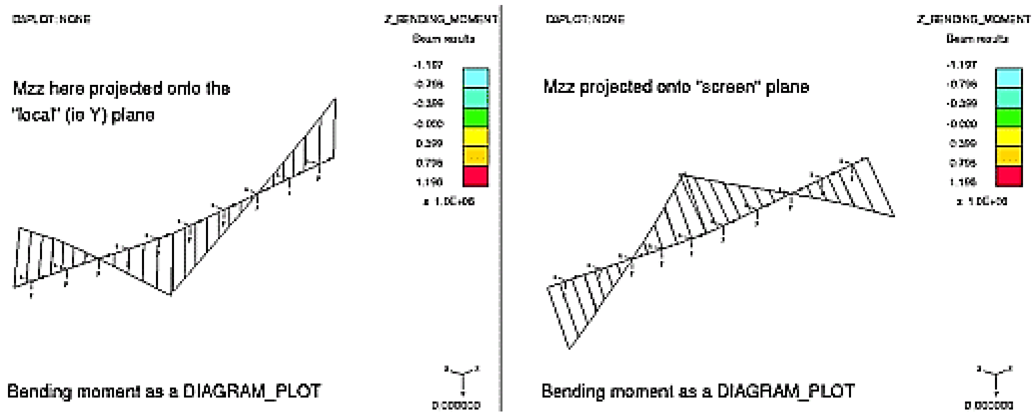
Note that "spotweld" (type 9 referencing **`*MAT_SPOTWELD`**) beams will only be drawn as such if a **`ZTF`** file for this analysis is found, as this contains the extra section data required to determine their attributes.

12.8.4. Diagram Plot Attributes

Diagram Plot Attributes

PROJECTION Using "local" or "screen" projection for diagram plots

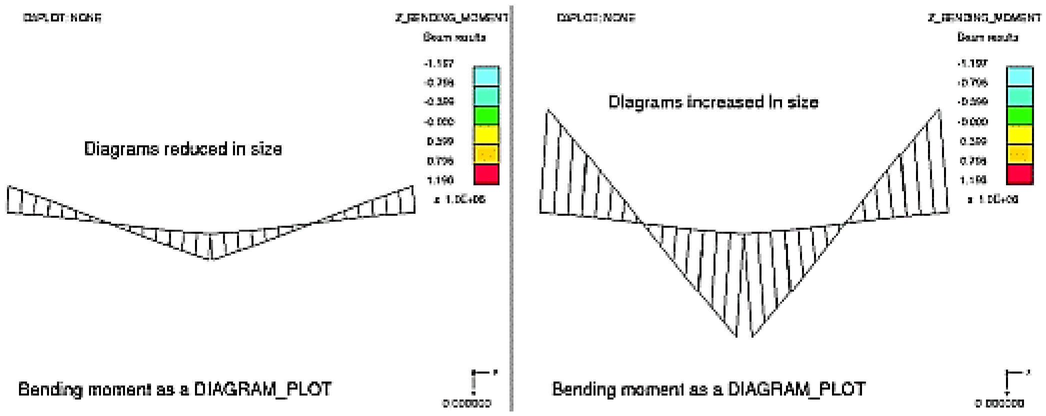
By default those data components which imply a local direction (i.e. shear and bending in/about beam local Y and Z axes) are drawn in "diagram" plots projected onto their "local" plane. For example in the left hand side of the figure above the bending moment **`Mzz`** is shown projected onto the plane of beam local XY. (The beam local axes have been turned on for clarity.) This gives a visual indication of the direction and sign of the component.



You can choose instead to project results onto the "screen" plane, as shown in the right hand side of the figure above. This draws results in the plane of the screen regardless of the beam's orientation. This mode of display is used unconditionally for components that do not have an implicit direction (e.g. axial force, torsion, etc)

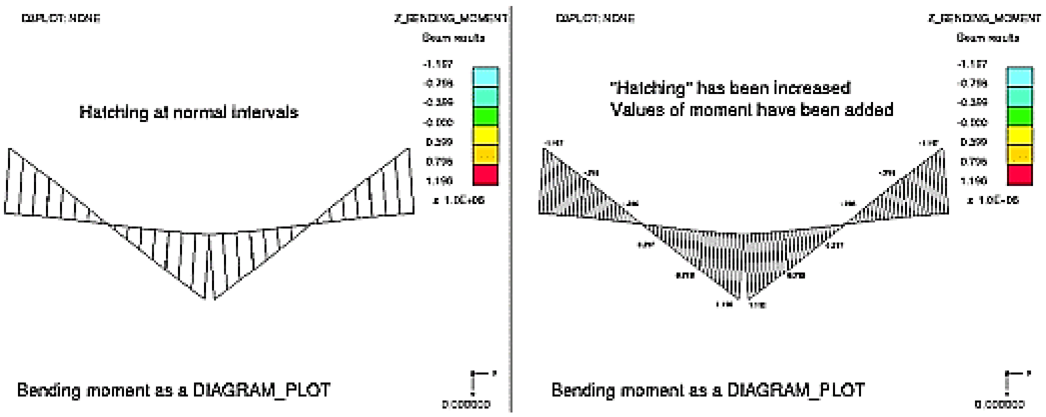
DIAGRAM SIZE Setting the visual scale of diagram plots

The size of diagram plots is set by default such that the largest vector is about 500 screen units. You can change this at will: the figure above shows typical settings.



HATCHING Setting the density of diagram plot hatching

The size of diagram plots is set by default such that the largest vector is about 500 screen units. You can change this at will: the figure above shows typical settings.



Note: A "solid" diagram plot appearance can be achieved by reducing the hatching spacing to typically 10 units or below. The actual value will depend on the screen window size and device resolution. It requires a lot of screen vectors to achieve this effect, so don't use it where display speed or storage space are critical: e.g. during animation.

12.8.5. Common Plotting Attributes

Common Plotting Attributes

LABEL VALUES Labelling values on diagram plots

The right hand figure above also shows the effect of turning on the **LABEL_VALUES** switch: values at beam end points are shown. This option is only useful when there are relatively few beams on the display, with a lot of beams being labelled the screen will become a mass of numbers. (The **LABEL** option, using "dynamic" labelling of beams with the **DATA_VALUE** switch on, provides a more selective way of drawing element data values on the screen.)

REVERSE_END_2 Sign convention used for beam plot display

The convention when drawing bending moment diagrams is to plot the diagram on the tensile side of the element. However the mathematics would suggest otherwise:

In the example used here (encastre beam, point load) the moments at the two ends of this beam are clockwise (+ve) and anti-clockwise (-ve) respectively, so at one end a +ve value must be drawn on the tensile side but at the other end a -ve value. Hence the need to reverse the sign of the "end 2" value for plotting purposes only. The author has never seen plots drawn without this reversal but, for completeness, the ability to produce them is provided with the **REVERSE_END_2** switch: turn it off to see mathematically "pure" plots.

IGNORE_BEAM_TYPE Ignore whether a beam is integrated or resultant when contouring integrated and resultant beam components

When this switch is off D3PLOT will use information in the ZTF file to determine whether a beam element is integrated or resultant type and will only plot corresponding integrated or resultant beam components on the beam elements. If the switch is on (or there is no ZTF file) then when user selects an integrated or resultant beam component all beams will be contoured with the value in the corresponding extra data slot, which will show inapplicable values on some beams. See Notes on Beam Data for more information.

12.8.6. WARNINGS On-Line Warnings About Beam-Plotting Pitfalls

WARNINGS On-line warnings about beam-plotting pitfalls

There are several different beam formulations in LS-DYNA, the principal ones being:

Hughes-Liu	Standard/arbitrary section, integrated at mid-span
Belytschko-Schwer	Standard sections, integrated at two ends
Truss	Axial only Belytschko-Schwer beam
Discrete	Zero-length, generalised spring-like behaviour
Cable	Tension only
Spotweld	A variation on discrete

Various data output options exist depending on material type, beam formulation, number of integration points and user-defined settings. Unfortunately the output files do not contain enough information to allow beam results to be diagnosed unambiguously, so **WARNINGS** provides on-line guidance. You should also read Beam element results which describes beam output.

Inconsistent sign conventions in LS-DYNA releases up to and including 970

Due to a bug in LS-DYNA versions up to and including LS970 exhibit the following inconsistent sign convention for beam output:

- "Resultant" (typically Belytschko-Schwer) elements use one sign convention
- "Integrated" (typically Hughes-Liu) elements use the opposite sign convention for 4 of the 6 output components.

The following table shows the results from releases 970 and earlier:

Component	Matching?
Fx	Same
Fy	Opposite
Fz	Opposite
Mxx	Opposite
Myy	Opposite
Mzz	Same

Which is right?

Sadly there is no "right" for beam output, as different users have different conventions. The confusion arises because of the different ways in which the beam types work: integrated beams have integration points at their centre, whereas resultant beams have (potential) hinges at their ends. The former reports force in the beam, and the latter reactions at the supports.

D3PLOT attempts to draw bending moment diagrams on the tensile side, but depending on which beam type you have used this may or may not be the case.

Sign conventions are consistent from LS-DYNA release 971 onwards

At some stage during the development of LS971 this problem was fixed, and results now use the "integrated" convention for all beam types.

Unfortunately D3PLOT cannot tell with certainty which LS-DYNA version was used to generate a set of results, so it cannot correct for this automatically. This has consequences for cut-section force extraction from beams, described in FORCES Computing Forces and Moments on the Cutting Plane

Update : From approximately 2009 onwards output from LS-DYNA 971 reports its version number correctly in output files, and D3PLOT is thus able to determine that the sign convention problem described above has been fixed.

Interpreting beam results requires knowledge about the original beam and material formulations that is not available in the database (`.ptf`) files. As a consequence there are some ambiguities in the interpretation of results, especially for "extra" data, that D3PLOT is unable to resolve for you.

It is your responsibility to determine what your results mean, and to interpret them correctly.

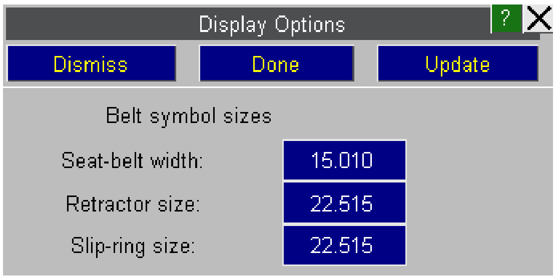
If you need advice or help please contact Oasys Ltd.

12.9. BELT_SYMBOLS... Menu: Setting the Sizes of Seat Belt and Related Symbols

BELT_SYMBOLS... menu: Setting the sizes of seat-belt and related symbols

You cannot change the symbols used for drawing seat-belts and related elements, but you can change their visual size in this menu.

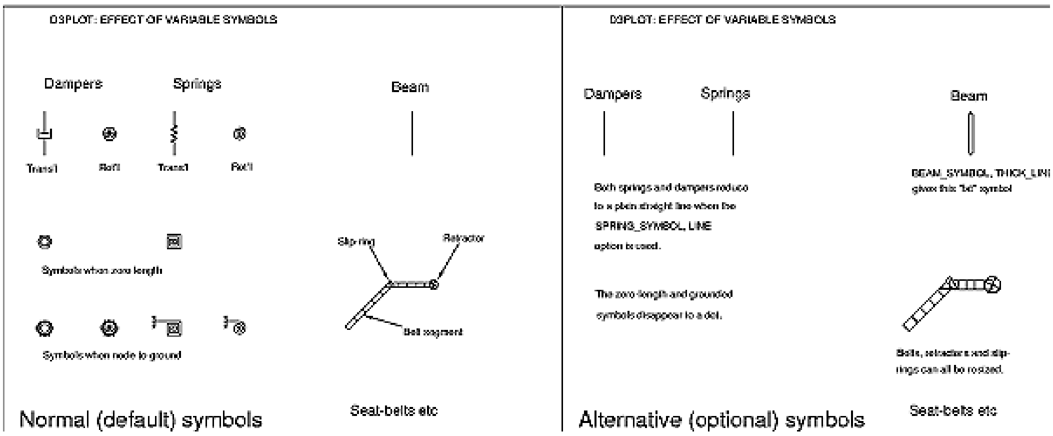
All dimensions are expressed in model space units and can be changed at will, the default values are only an estimate of what should look sensible.



12.9.1. Summary of Default and Modifiable Symbols of Springs, Beams and Seatbelts

Summary of default and modifiable symbols of Springs, Beams and Seatbelts

The figure below left shows default symbols for all springs, beams and belt types; and the figure below right shows what they can be modified to become.

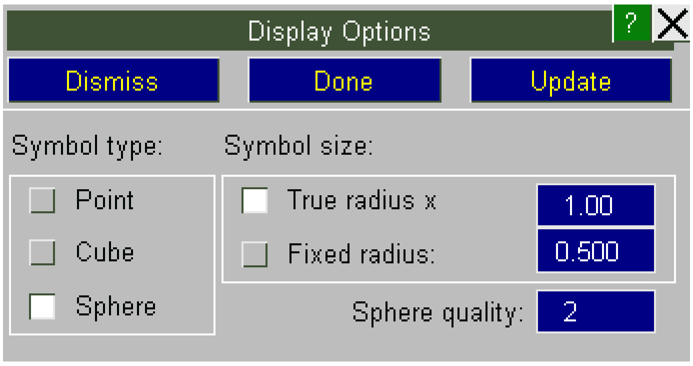


12.10. SPH Symbols Managing SPH Element Display

SPH Symbols. Managing SPH element display

Smooth Particle Hydrodynamic (SPH) elements are spheres with a given "radius of influence" that can change over time.

By default D3PLOT renders them as spheres of the current diameter, treated as opaque for shading and hidden surface removal, but this can be changed using the options here.



12.10.1. Symbol Type: Controlling the Method Used to Draw SPH Elements

Symbol type: controlling the method used to draw SPH elements

Point Uses "points" to display elements. These are 2 dimensional squares drawn in the plane of the screen at the appropriate location, and with width and height "radius" x 2.

OpenGL renders "points" extremely fast, meaning that this is an efficient display method if you have many SPH elements; however the symbol has no depth, is not "lit", and there will be a hardware-specific maximum size for a point symbol so it may not show the true size of elements.

Cube Uses a cube of width, height and depth "radius" x 2 to display elements.

This only ever shows 3 faces, so it is reasonably fast to draw, but it looks a bit odd showing a spherical element as a cube. However cubes have depth and orientation, and can be lit, so the result is better-looking that a "point".

Sphere Draws a sphere of the relevant radius, giving a "true" element appearance. However spheres require many facets for rendering, making these slower to draw.

Sphere quality is a value between 1 and 5 which determines the number of facets used to render the sphere symbols. Each increment halves the equatorial and meridional angular increment size, quadrupling the number of facets on the symbol.

The default value of 2 gives a good compromise between symbol quality and rendering speed, but you may wish to increase this figure when generating presentation quality images.

This image shows a "chicken" made of SPH elements striking a flat plate, and demonstrates the default spherical symbols at quality level 2 and their true radius. The SPH elements have been contoured with their radius.

12.10.2. Symbol Size: Controlling SPH Symbol Display

Symbol size: controlling SPH symbol display

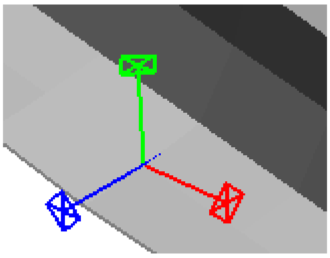
True radius x < *factor* > uses the radius of influence reported for the current state multiplied by <factor>. This is the default and is recommended for most cases.

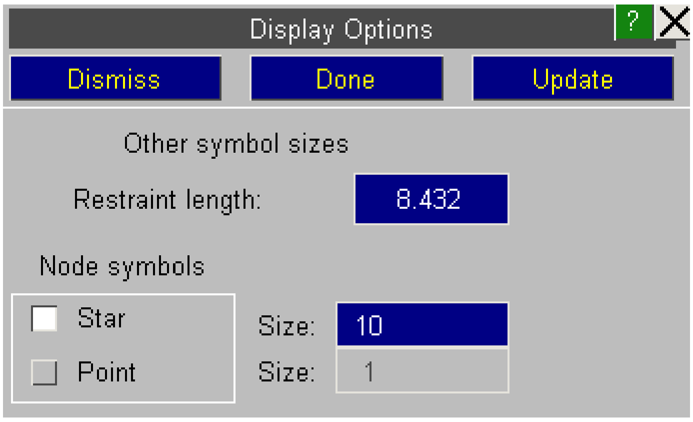
Fixed radius < *value* > uses the stipulated fixed radius size. This can sometimes be useful if analyses have gone wrong somehow and delivered very large (or small) radii, making it impossible to see what is happening unless the symbol size is set explicitly.

12.11. Other Symbols

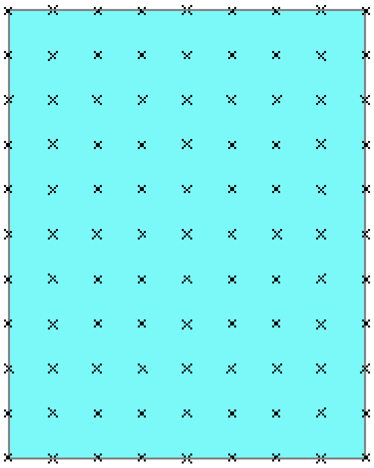
Other Symbols

Restraint length is the size given to restraint and constraint symbols applied to nodes in model space units.





Node symbols controls how nodes are shown. Normally they use a star, but these can be quite slow to draw on some hardware, and points will render many (up to 100x) faster.



This image shows a panel of shells with the default "star" symbol seen head on. The images on the right enlarge a corner detail to show both "star" and the "point" alternative.

This image shows "stars"

This image shows "points"

12.12. AB Pcle Symbols: Managing Airbag Particle Display

AB Pcle Symbols: Managing Airbag Particle display

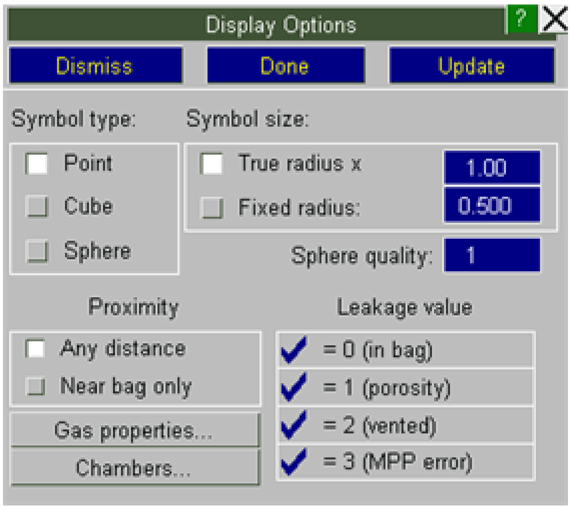
Airbag particles can be thought of as small spheres of gas emitted from an inflator to apply pressure inside an airbag. D3PLOT treats them as "elements" for display and contouring purposes.

They are rendered using symbols that are the same as those used for SPH elements. However the two element types are highly unlikely to appear in the same analysis, and their relative sizes are also likely to be very different, so despite the potential for confusion this similarity is not a problem.

Airbag particles start off "latent" at the beginning of an analysis, and then burst into life as the analysis proceeds and the inflator releases them into the bag. An inflator may have several separate "gases", and particles will belong to a particular "gas".

Once "alive" particles will start off inside the bag, but they may escape from it either by migrating through porous fabric or by traveling through a vent hole, and their "leakage" value will change to reflect this status.

In addition particles may be "near" to the bag fabric, and hence exerting pressure upon it, or "distant" and not affecting the fabric directly.



12.12.1. Symbol Type: Controlling the Method Used to Draw ABP Elements

Symbol type: controlling the method used to draw ABP elements

Three different symbol types are provided for displaying airbag particles giving a trade-off between image quality and rendering speed.

Point

Uses "points" to display particles. These are 2 dimensional squares drawn in the plane of the screen at the appropriate location, and with width and height "radius" x 2.

"Points" are drawn extremely fast in OpenGL, and since a typical airbag may have tens of thousands of particles this is the default display method.

Cube

Uses a cube of width, height and depth "radius" x 2 to display particles.

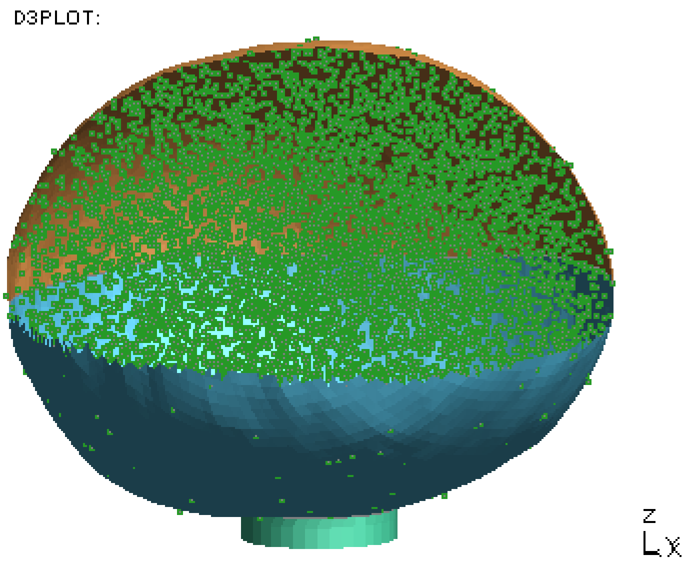
This only ever shows 3 faces, so it is reasonably fast to draw, but it looks a bit odd showing a spherical element as a cube. However cubes have depth and orientation, and can be lit, so the result is better-looking that a "point".

Sphere

Draws a sphere of the relevant radius, giving a "true" particle appearance. However spheres require many facets for rendering, making these slower to draw.

Sphere quality is a value between 1 and 5 which determines the number of facets used to render the sphere symbols. Each increment halves the equatorial and meridional angular increment size, quadrupling the number of facets on the symbol.

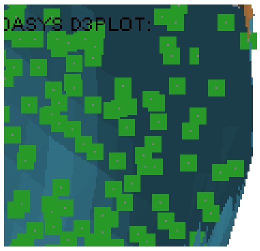
The default value of 1 gives a rather "pointy" looking particle symbol, but it is usually acceptable given their small size. Higher quality values may be necessary when generating images for presentation.



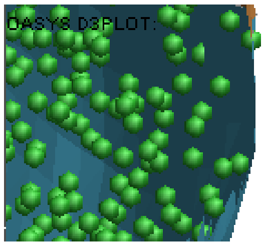
This image shows a typical "pancake" airbag being inflated by particles. Part of the fabric has been cut away to show the particle elements inside which are drawn by the default "point" method.

The three images on the right show a detail of the right hand side to demonstrate how the different display methods appear when enlarged. For this particular analysis the "point" display method gave an animation speed that was limited by the refresh rate of the monitor at 85 frames per second, whereas the "sphere" methods displayed at about 30 fps or slower.

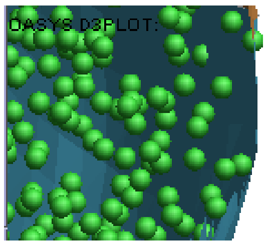
Default "point" method.



Spheres, quality = 1



Spheres, quality = 2

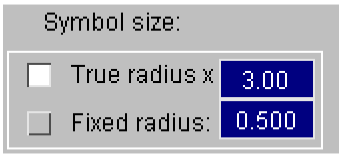


12.12.2. Symbol Size: Controlling ABP Symbol Display

Symbol size: controlling ABP symbol display

True radius x < *factor* > uses the radius reported for the current state multiplied by <factor>. This is the default and is recommended for most cases. In all the analyses encountered so far particle radius has remained constant throughout the run, however it is written as a value at every state so this may change in the future.

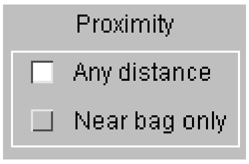
Fixed radius < *value* > uses the stipulated fixed radius size.



12.12.3. Proximity: Limiting Display to Particles "Near" the Bag Fabric

Proximity: limiting display to particles "near" the bag fabric

By default all "live" particles in the bag are shown, but only those that are "near" to the bag fabric will be exerting pressure on it, and display can be limited to these particles only to give a less cluttered result.



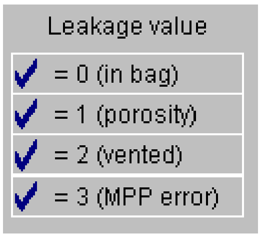
12.12.4. Leakage Value: Limiting Display to Particles Inside or Outside the Bag

Leakage Value: limiting display to particles inside or outside the bag

Once "alive" particles will have one of the following leakage values:

0. Inside the bag
1. Escaped due to fabric porosity
2. Escaped through a vent hole
3. MPP error

Display may be restricted to any permutation of these. (By default all are shown.)



12.12.5. Gas Properties: Setting Display Attributes for Individual Gases

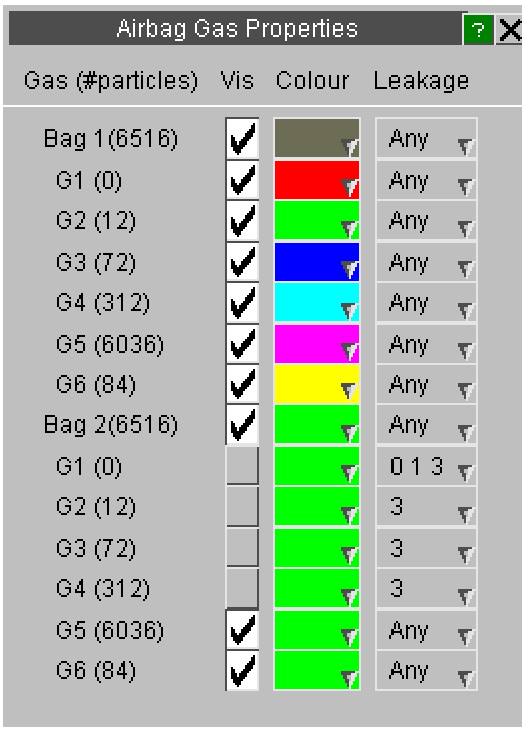
Gas properties: setting display attributes for individual gases

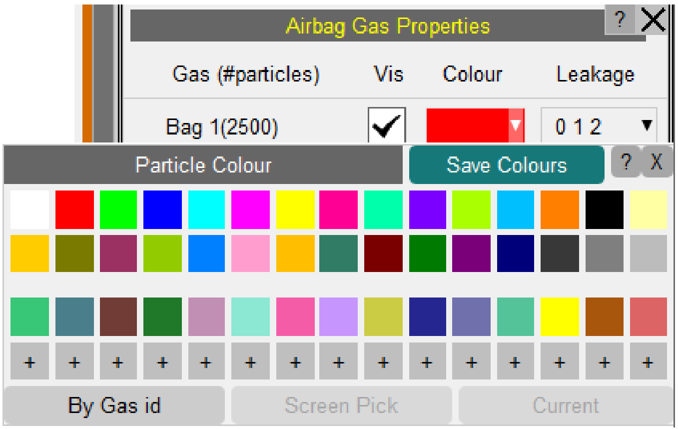
By default the settings above apply to all particles in all airbags in the active windows for the panel, and this is generally sufficient. However Particle Airbags may have several "gases" (NGAS on the input card) and it is possible to set display attributes on a per-gas basis.

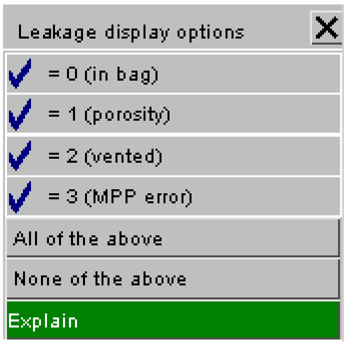
Each airbag in the model will be listed together with the gases in the bag.

Selecting an attribute for the bag as a whole will propagate the selection down to all its gases, selecting attributes for an individual gas will only affect that gas.

The following attributes are controllable:



Gas id and #particles	The number of particles "live" at the state current when the panel was mapped is shown. This figure is not updated automatically as states change, but may be updated to the current state at any time by clicking on the **Gas Properties** button again.
Vis(*ibility* **)**	Whether or not the particles in this gas will be displayed. By default all will be shown, but any gas may be switched on/off individually. Note that this is not the same as blanking particles, rather it is analogous to turning the "entity display switch" for the gas on or off.
Colour	Selects the colour for the particles in this gas. By default all particles in a bag will inherit the colour of the airbag itself (as set via the **Properties** or **Colour** panels), but individual colours may be specified for each gas within a bag. The colour option " **By Gas id** " may be used at the Bag level to set a range of colours for all gases in the bag automatically. 

Leakage	Selects display based on leakage values. By default all leakage states are shown, but any permutation of values may be set. Note that selecting a global leakage value in the main options panel (as described [above](#)) will propagate down to all active bags, overwriting any individual settings made here. 

12.13. Spotweld Symbols: Managing Spotweld Element Display

Spotweld Symbols: Managing Spotweld element display

D3PLOT contains a number of different options for controlling the symbol and the size of the symbol used to display spotwelds.

This option controls the size of the symbols used to draw some spotwelds. D3PLOT 9.4 can draw and contour five different types of spotweld

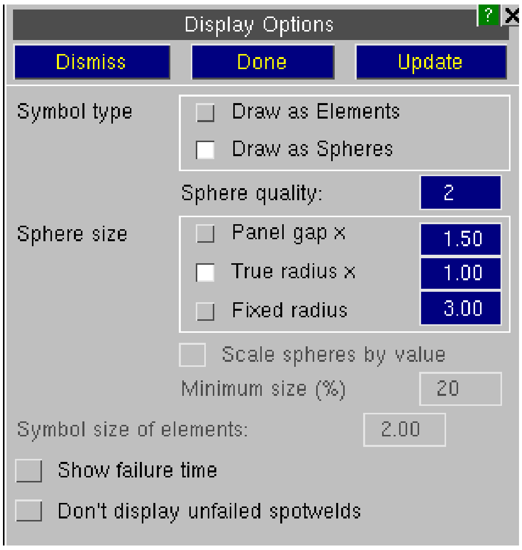
Symbol Type

Draw as Elements

Spotwelds may be drawn using the symbol types introduced in version 9.4 where each type of spotweld has a different symbol as shown below. For spotweld beams, solids and clusters this means they are drawn using the solid and beam element geometry.

Draw as Spheres

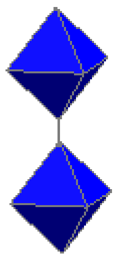
From version 13.0 onwards by default spotwelds are drawn as spheres located at the center of each spotweld.



12.13.1. *CONSTRAINED_SPOTWELD

*CONSTRAINED_SPOTWELD

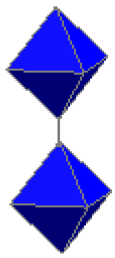
These are drawn and contoured as two diamonds connected together by a line. They are labeled as **CWn** .



12.13.2. *CONSTRAINED_GENERALIZED_WELD_...

*CONSTRAINED_GENERALIZED_WELD_...

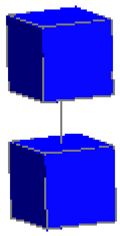
These are drawn and contoured as two diamonds connected together by a line. They are labeled as **GWn** .



12.13.3. *MAT_SPOTWELD (Beams)

*MAT_SPOTWELD (beams)

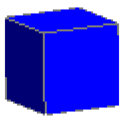
These are drawn and contoured as two cubes connected together by a line. They are labeled as **BWn** .



12.13.4. *MAT_SPOTWELD (Solids)

*MAT_SPOTWELD (solids)

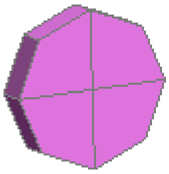
These are drawn and contoured using the solid element that defined the spotweld. They are labeled as **HWn** .



12.13.5. *DEFINE_HEX_SPOTWELD_ASSEMBLY

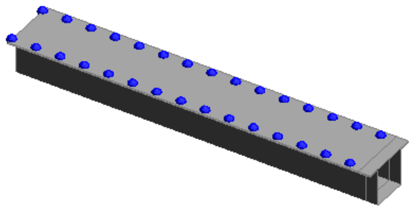
*DEFINE_HEX_SPOTWELD_ASSEMBLY

These are drawn and contoured using the solid elements that define the spotweld assembly. All of the solid elements are contoured using the same colour as the LSDA file contains a single value for each assembly. They are labeled as **HAn** .



Spheres

In some plots where the spotwelds lay between panels and can not be seen using the version 94 symbols swapping to spheres which protrude through the panels can allow the spotwelds to be seen.



Sphere Size

To make it easier to view the spotweld locations the size of the spheres can be controlled using a number of different options.

Panel Gap x < *factor* >

D3PLOT will calculate a panel gap based on the initial geometry of each weld.

CONSTRAINED_SPOTWELD	0.5 x the distance between the 2 nodes
CONSTRAINED_GENERALIZED_WELD_	0.5 x the distance between the 2 nodes
MAT_SPOTWELD (beams)	0.5 x the distance between the 2 nodes

MAT_SPOTWELD (solids)	0.5 x the average of the distance between the 4 pairs of of nodes that make up the edges of the spotweld.
DEFINE_HEX_SPOTWELD_ASSEMBLY	0.5 x the average of the distance between all of the pairs of nodes that make up the edges of the solids.

True Radius x < *factor* >

This uses the radius of each spotweld. For each spotweld type D3PLOT will use the following for each radius.

CONSTRAINED_SPOTWELD	0.5 x the distance between the 2 nodes
CONSTRAINED_GENERALIZED_WELD_	0.5 x the distance between the 2 nodes
MAT_SPOTWELD (beams)	0.5 x beam cross section diameter
MAT_SPOTWELD (solids)	If the weld was created as a PRIMER connection then 0.5 x the connection diameter will be used. If no connection data is available then D3PLOT will calculate a radius based on the solid geometry.
DEFINE_HEX_SPOTWELD_ASSEMBLY	D3PLOT will calculate a radius based on the solid elements geometry.

Fixed radius < *value* >

This uses the stipulated fixed radius size. This can sometimes be useful if some of the spotwelds are very small and can not easily be seen.

Scale symbols by value

If a **Fixed radius** is used then the fixed radius can be scaled according to the magnitude of the data value when spotweld data components are contoured.

Minimum size (%)

If a **Fixed radius** is used then this option can be used to set a lower limit when the size is scaled by value.

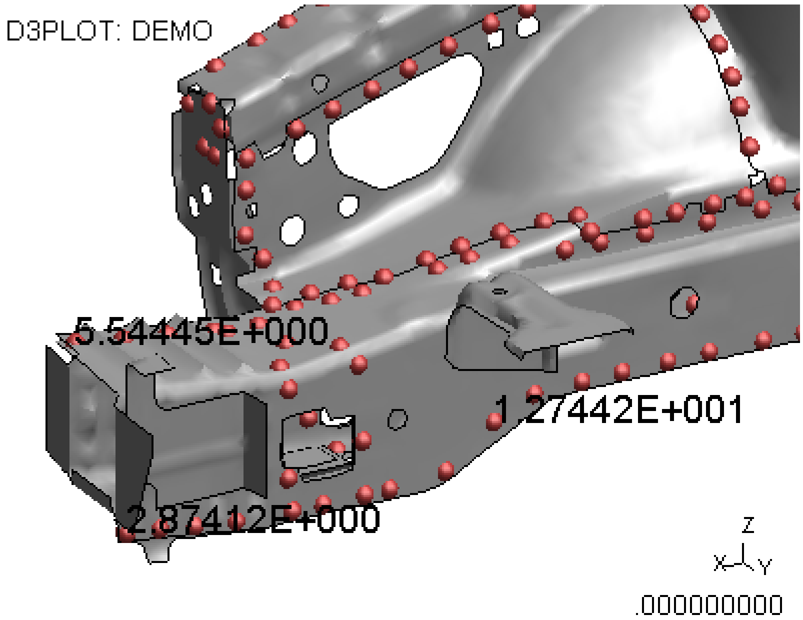
Version 94 symbols size

This option can be used to scale the size of the version 9.4 symbols.

Show Failure Time

This option will automatically annotate any spotwelds that fail during the analysis with the failure time. Only spotwelds with a failure time greater than 0.0 are displayed.

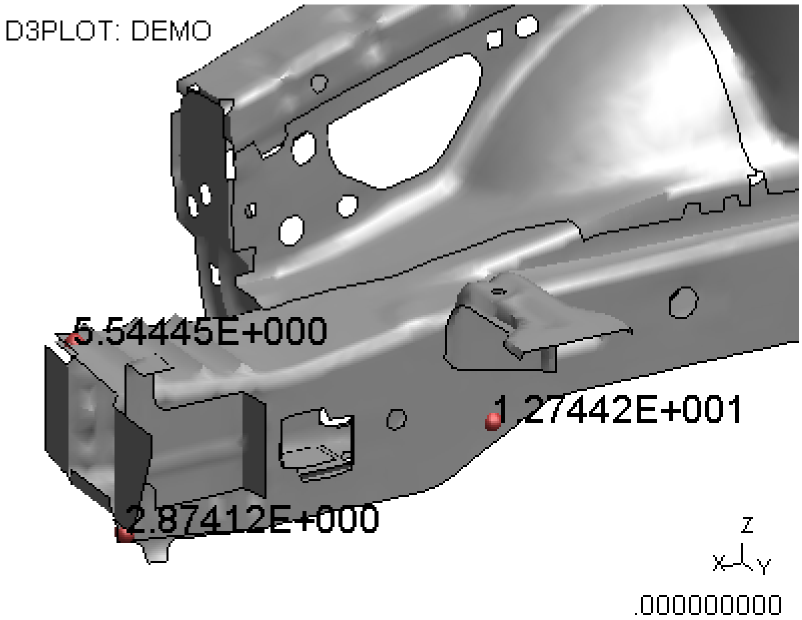
The failure times displayed are taken from the last state in the LSDA (binout) file and are constant regardless of the plot state time.



Don't display unfailed spotwelds

To make it easier to identify the spotwelds that fail this option can be used to automatically turn off the display of any spotwelds that do not fail during the analysis

As with the display of the failure time the information from the last state in the LSDA (binout) file is used to determine which spotwelds fail.



Spotweld Preference Options

The following preference options can be used to set the default options used to display spotwelds

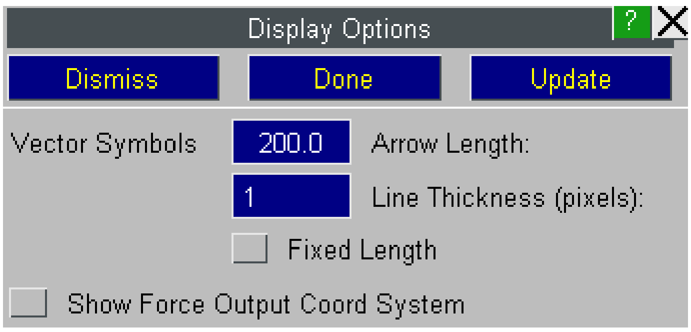
d3plot*swld_symbol	Symbol for type for Spotwelds, either DEFAULT or SPHERE
d3plot*swld_quality	Quality of Spotweld sphere symbol (1-5)
d3plot*swld_radius	Display spotwelds using the PANEL gap, TRUE radius or a FIXED radius (PANEL, TRUE, FIXED)
d3plot*swld_panel_factor	Factor to multiply PANEL gap by when drawing spotwelds spheres
d3plot*swld_true_factor	Factor to multiply TRUE radius by when drawing spotwelds spheres
d3plot*swld_fixed_size	Default radius used when drawing spotwelds with a FIXED radius
d3plot*swld_scale_by_value	TRUE if spotweld radius is going to be scaled by the value

12.14. X-Section Symbols

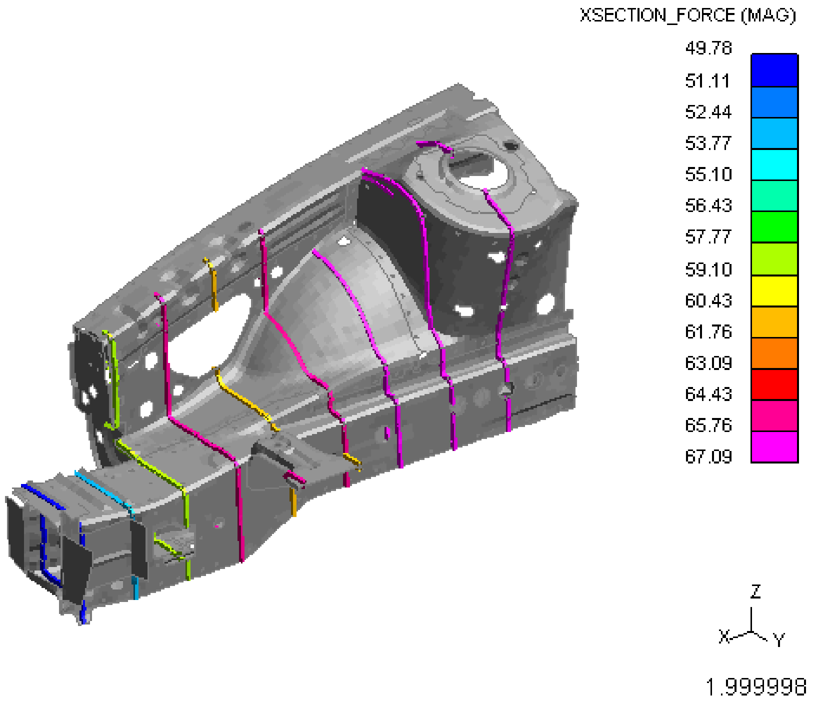
X-Section Symbols

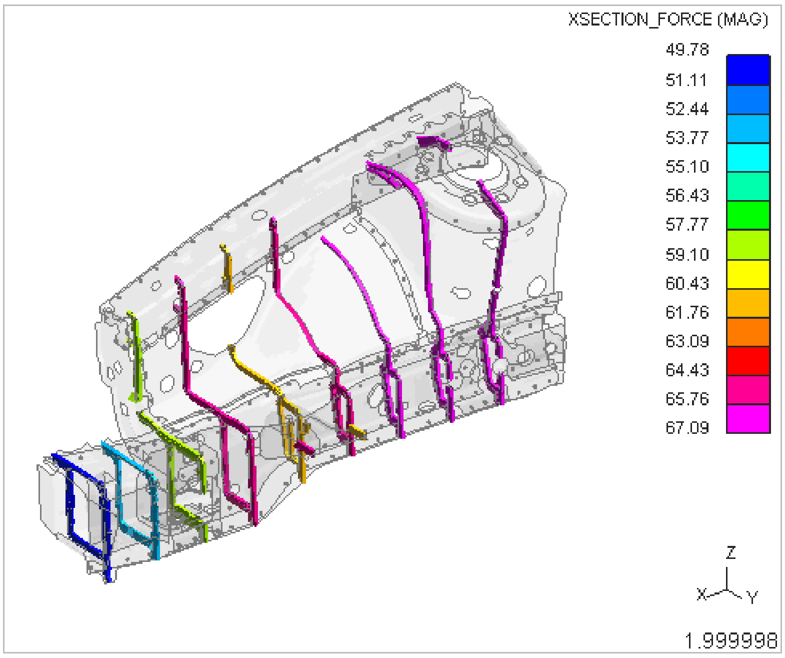
From version 10.0 onwards of D3PLOT can draw the location of any *DATABASE_CROSS_SECTION definitions defined in the LS-DYNA model model (requires a ZTF file generated by PRIMER).

As well as drawing the location of these cross sections D3PLOT can also contour force and moment results on them and generate force vector plots. These options can be used to control the size of the arrows used when generating vector plots.



As well as drawing the location of cross sections D3PLOT can also contour force and moment results on them and generate force vector plots

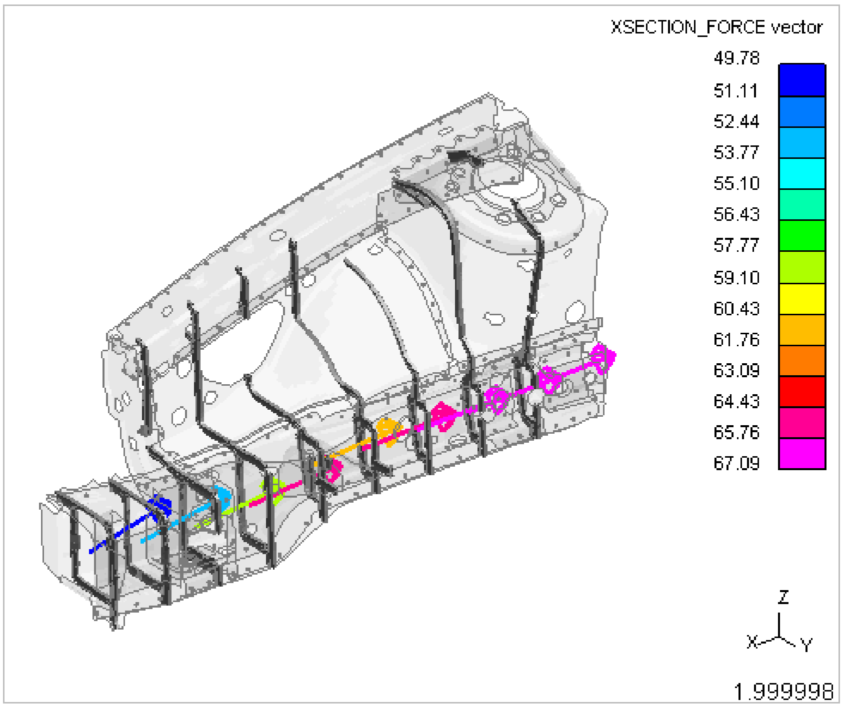




Vector Symbols

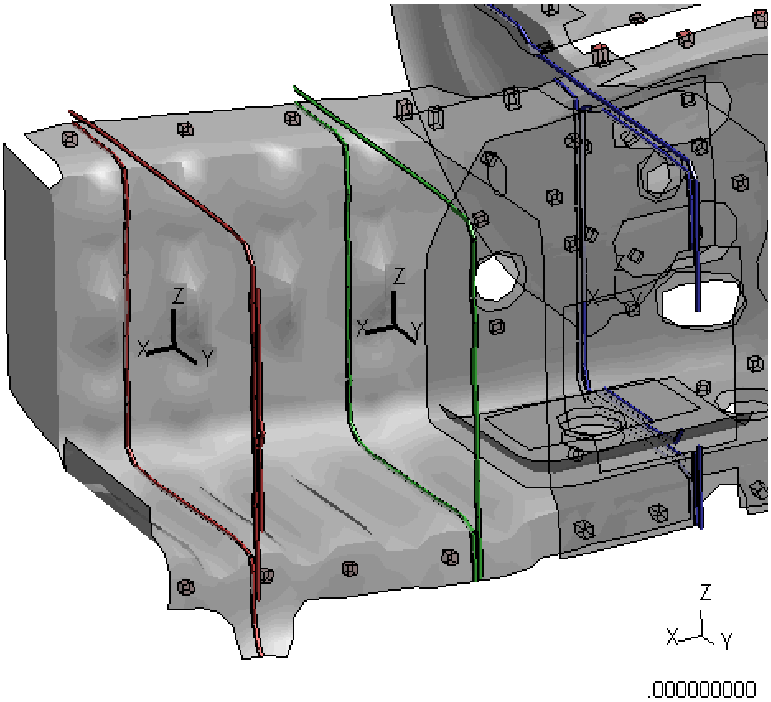
These options can be used to control the size of the arrows used when generating vector plots of SPC forces

Arrow Length	This is the maximum length used to draw the arrows used for vector plots.
Line Thickness	This option can be used to increase the width of the lines used to draw the arrows.
Fixed Length	By default the length of each arrow is scaled by the magnitude of the data value. If this option is selected then all the arrows will be drawn the same length and just the colour of the arrow will be used to indicate the value. This option can be useful if there is a large variation between values and the arrows for the smaller values are difficult to see.



Show Force Output Coordinate System

By default the forces for a



*DATABASE_CROSS_SECTION are written out using the global Cartesian coordinate system but this can be modified by changing the parameters ID and ITYPE on the *DATABASE_CROSS_SECTION card.

This option can be used to display a triad at the cross section centroid which shows the coordinate system for force output.

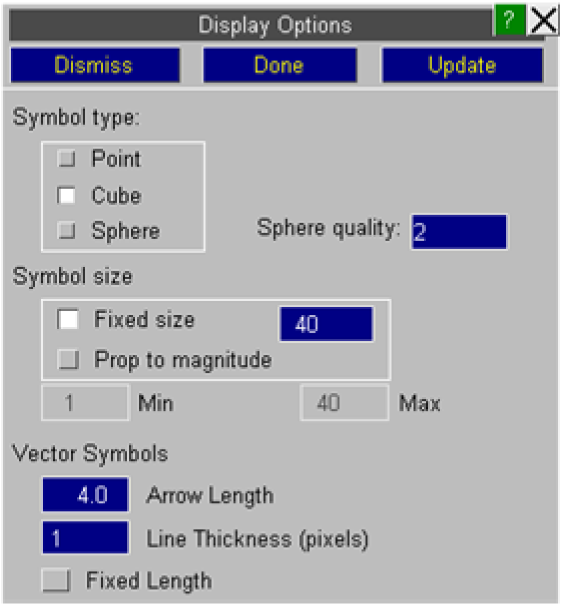
Using information from the ZTF file D3PLOT will update the triad if the local coordinate system is defined using either an accelerometer or a rigid body which moves during the analysis.

12.15. SPC Symbols

SPC Symbols

These option controls the size of the symbols used to draw and contour SPC's.

Three different symbol types are provided for displaying airbag particles giving a trade-off between image quality and rendering speed.



Point

These are 2 dimensional squares drawn in the plane of the screen.

"Points" are drawn extremely fast in OpenGL, and since a typical airbag may have tens of thousands of particles this is the default display method.

Cube

This only ever shows 3 faces, so it is reasonably fast to draw, but it looks a bit odd showing a spherical element as a cube. However cubes have depth and orientation, and can be lit, so the result is better-looking that a "point".

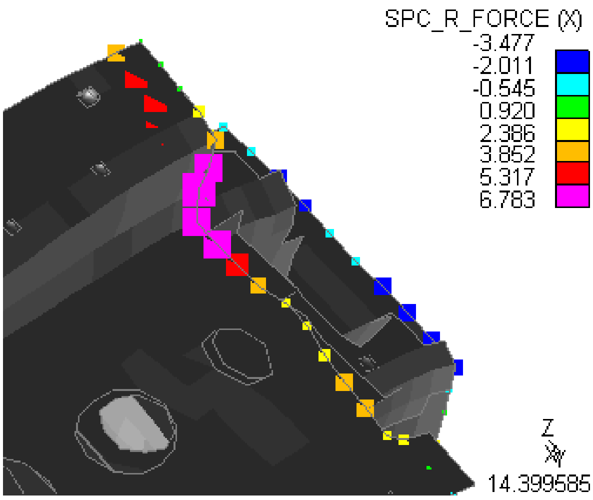
Sphere

Spheres require many facets for rendering, making these slower to draw.

Sphere quality is a value between 1 and 5 which determines the number of facets used to render the sphere symbols. Each increment halves the equatorial and meridional angular increment size, quadrupling the number of facets on the symbol.

The default value is 2.

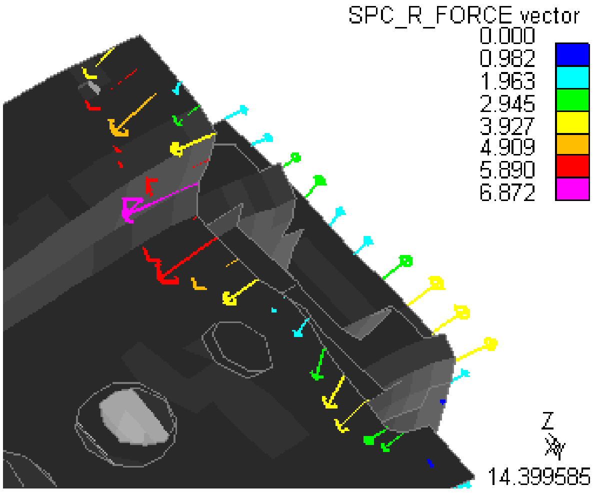
Fixed Size	All of the SPC symbols are drawn using the same size
Prop to magnitude	This option can be used to automatically scale the size of the SPC symbols in proportion to the magnitude of the data value.
Min (pixels)	This specified the minimum size used when scaling SPC symbols in proportion to the magnitude of the data value.
Max (pixels)	This specified the maximum size used when scaling SPC symbols in proportion to the magnitude of the data value.



These options can be used to control the size of the arrows used when generating vector plots of SPC forces.

Arrow Length	This is the maximum length used to draw the arrows used for vector plots.
Line Thickness	This option can be used to increase the width of the lines used to draw the arrows.
Fixed Length	By default the length of each arrow is scaled by the magnitude of the data value. If this option is selected then all the arrows will be drawn the same length and just the colour of the arrow will be used to indicate the value. This option can be useful if there is a large variation between values and the arrows for the smaller values are difficult to see.

Changing any of these values will affect all vector plots (e.g. Velocity)



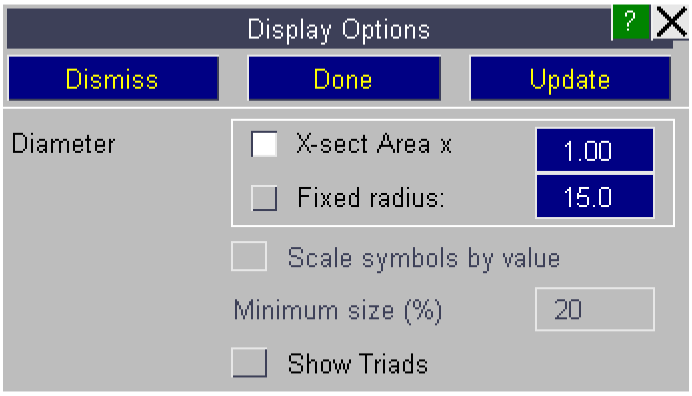
12.16. Load Paths

Load Paths

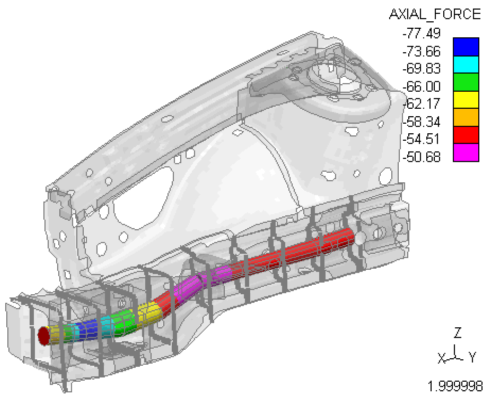
From D3PLOT 11.0 onwards, you can draw LOADPATHs joining the centres of *DATABASE_CROSS_SECTION definitions. These are created in PRIMER and require a ZTF file be generated.

The forces calculated from the *DATABASE_CROSS_SECTIONs can be plotted on the LOADPATHs to make visualisation of loads through a structure easier.

These options can be used to control the size of the LOADPATHs used when generating plots.

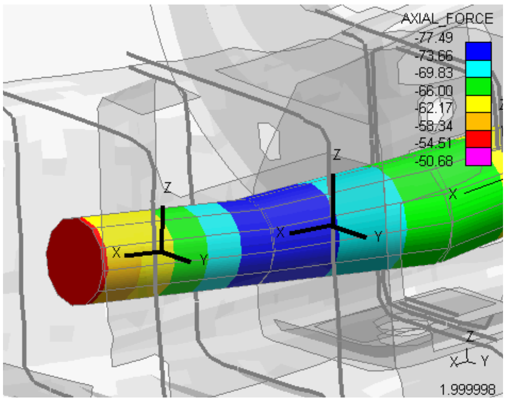


Diameter By default the diameter of each end of a LOADPATH segment is scaled based on the *DATABASE_ CROSS_SECTION cross section area. Selecting the **Fixed Radius** option will set their diameters to the same size.



Scale By Value If the **Fixed Radius** option has been selected then this option can be used to scale the diameters of each LOADPATH segment by it's current data value.

Show Triads This will show the local coordinate systems of each LOADPATH segment.

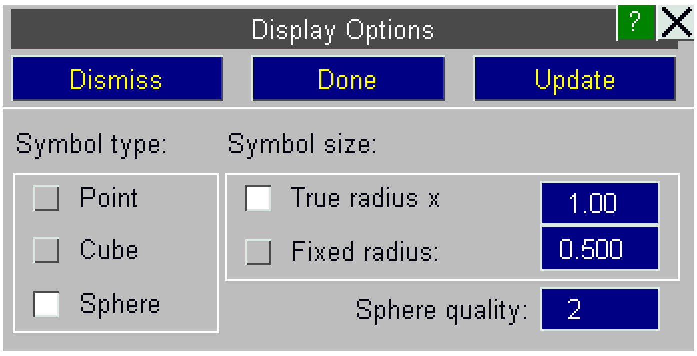


12.17. DES Symbols

DES Symbols

DES elements (*ELEMENT_DISCRETE_SPHERE) are used for discrete element calculations. Each particle consists of a single node with its mass, mass moment of inertia, and radius.

DES elements can appear during the analysis and are rendered using symbols that are similar to those used for SPH elements and Airbag Particles.



12.17.1. Symbol Type

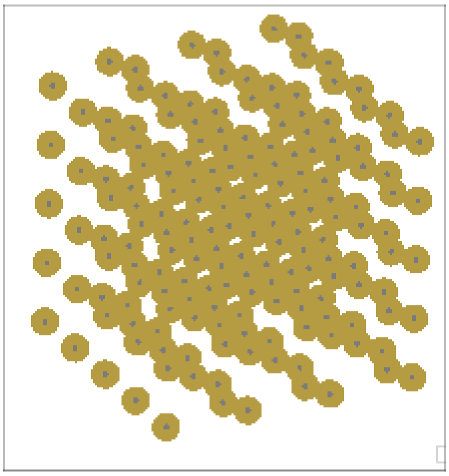
Symbol type

Three different symbol types are provided for displaying DES elements giving a trade-off between image quality and rendering speed.

Point

Uses "points" to display particles. These are 2 dimensional squares drawn in the plane of the screen at the appropriate location, and with width and height "radius" x 2.

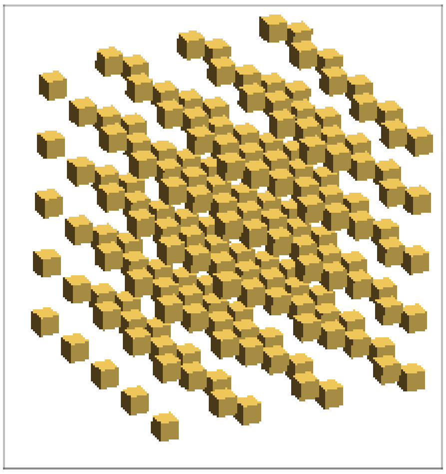
"Points" are drawn extremely fast in OpenGL, and since a typical analysis may have thousands of DES elements this is the default display method.



Cube

Uses a cube of width, height and depth "radius" x 2 to display particles.

This only ever shows 3 faces, so it is reasonably fast to draw, but it looks a bit odd showing a spherical element as a cube. However cubes have depth and orientation, and can be lit, so the result is better-looking that a "point".



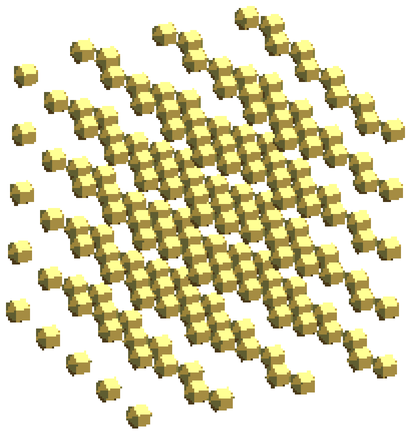
Sphere

Draws a sphere of the relevant radius, giving a "true" particle appearance. However spheres require many facets for rendering, making these slower to draw.

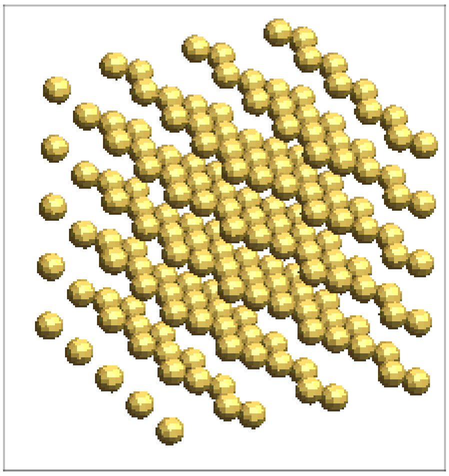
Sphere quality is a value between 1 and 5 which determines the number of facets used to render the sphere symbols. Each increment halves the equatorial and meridional angular increment size, quadrupling the number of facets on the symbol.

The default value of 1 gives a rather "pointy" looking particle symbol, but it is usually acceptable given their small size. Higher quality values may be necessary when generating images for presentation.

Spheres, quality = 1



Spheres, quality = 2

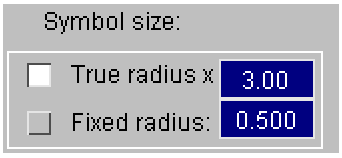


12.17.2. Symbol Size: Controlling DES Symbol Display

Symbol size: controlling DES symbol display

True radius x < *factor* > uses the radius reported for the current state multiplied by <factor>. This is the default and is recommended for most cases. In all the analyses encountered so far particle radius has remained constant throughout the run, however it is written as a value at every state so this may change in the future.

Fixed radius < *value* > uses the stipulated fixed radius size.

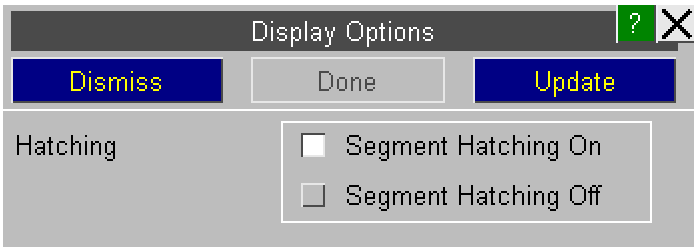


12.18. Interface Symbols

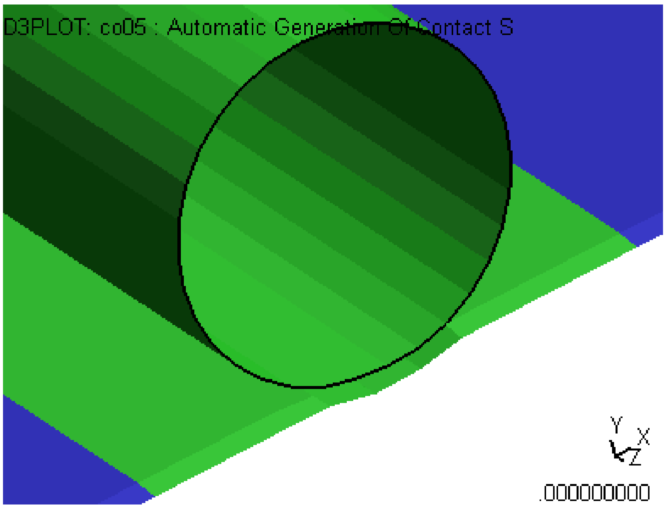
Interface Symbols

Contact surface interface segments are coincident with the faces of 2D and 3D elements. This can make it very difficult to see when the segments are located.

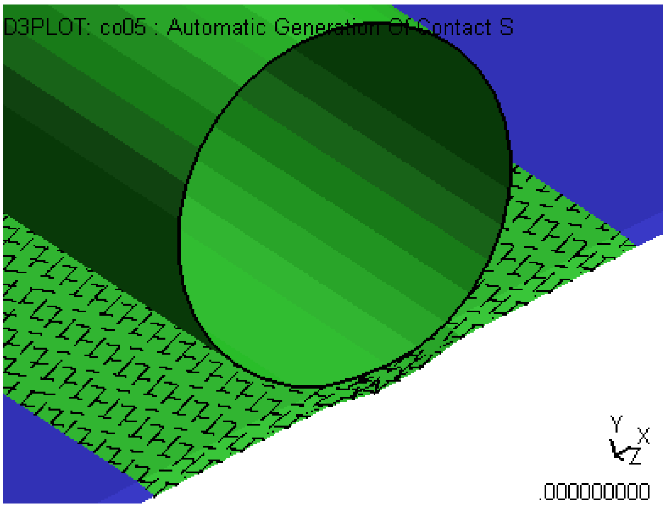
To make it easier to see the contact segments a cross hatching can be drawn on top of each segment.



Segment Hatching Off



Segment Hatching On

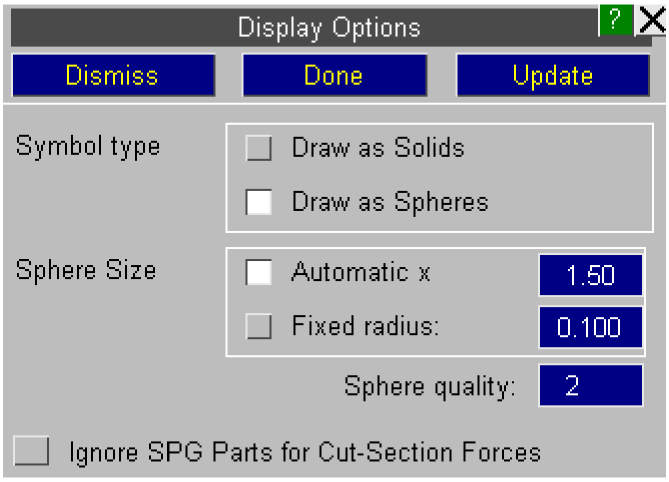


12.19. Solid SPG Symbols

Solid SPG Symbols

This option controls how Solid SPG (Smoothed Particle Galerkin) parts are drawn.

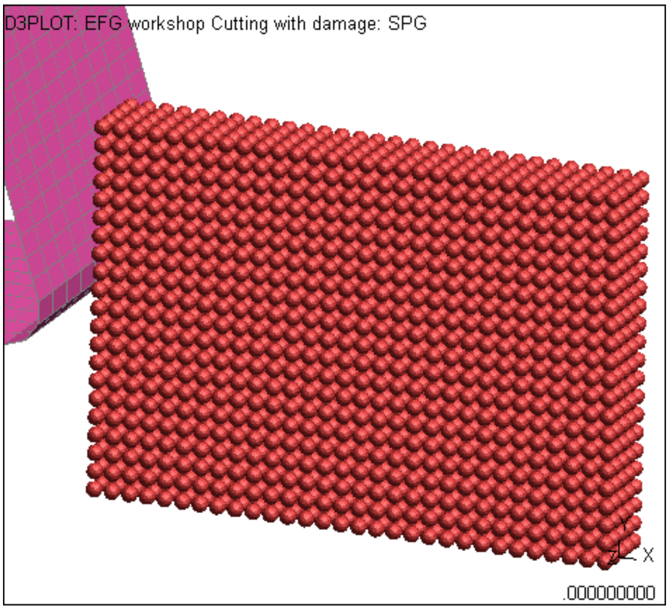
If D3PLOT reads a ZTF then it can automatically identify by parts have been defined using *SECTION_SOLID_SPG and by default it will display those parts by drawing spheres at the nodes instaed of using the normal solid element mesh.

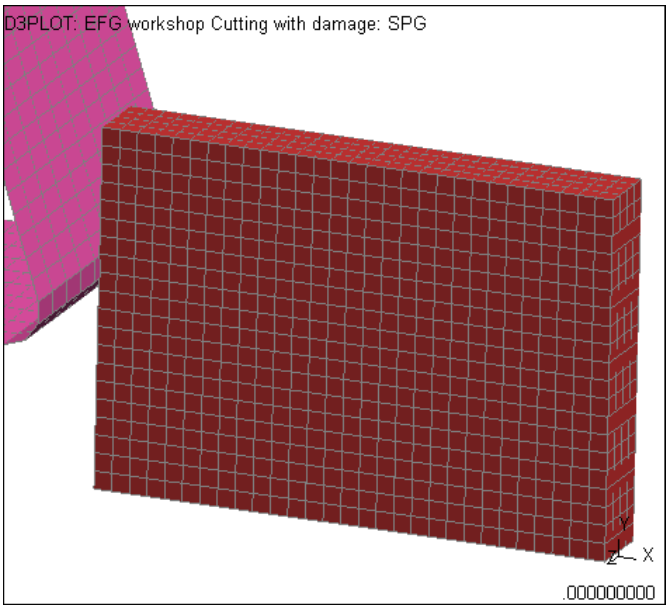


12.19.1. Symbol Type

Symbol Type

By default D3PLOT will draw SPG parts using spheres. This option can be used to swap the drawing mode of the SPG parts back to normal solid element.



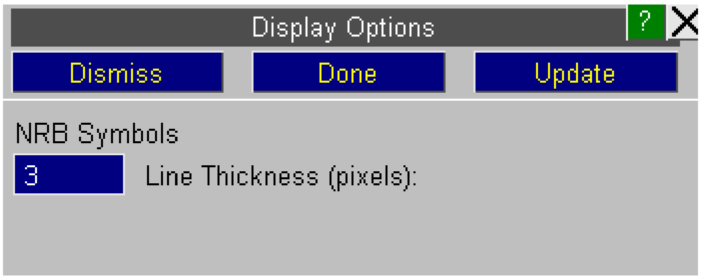


12.20. NRB

NRB

Line Thickness

By default Nodal Rigid Bodies will be drawn using a line thickness of 3 pixels. This thicknesscan be changed to anything between 1 and 10 pixels

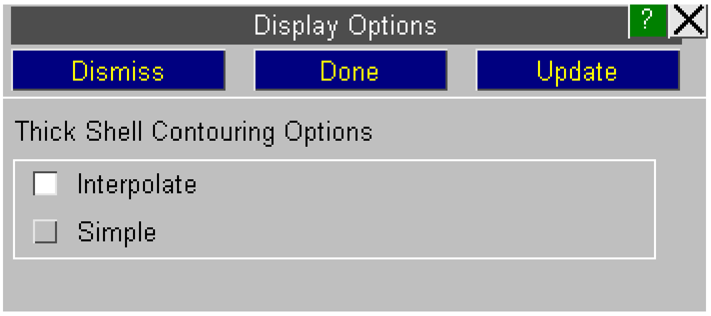


12.21. Thick Shells

Thick Shells

By default Thick Shells (*ELEMENT_THSELL) are contoured using the the topmost (outer) integration point for the top surface and the bottommost (inner) integration point for the bottom surface. The data on the side faces is then interpolated between these 2 values. See Contour Display Options for Thick Shell elements for more details about this interpolation.

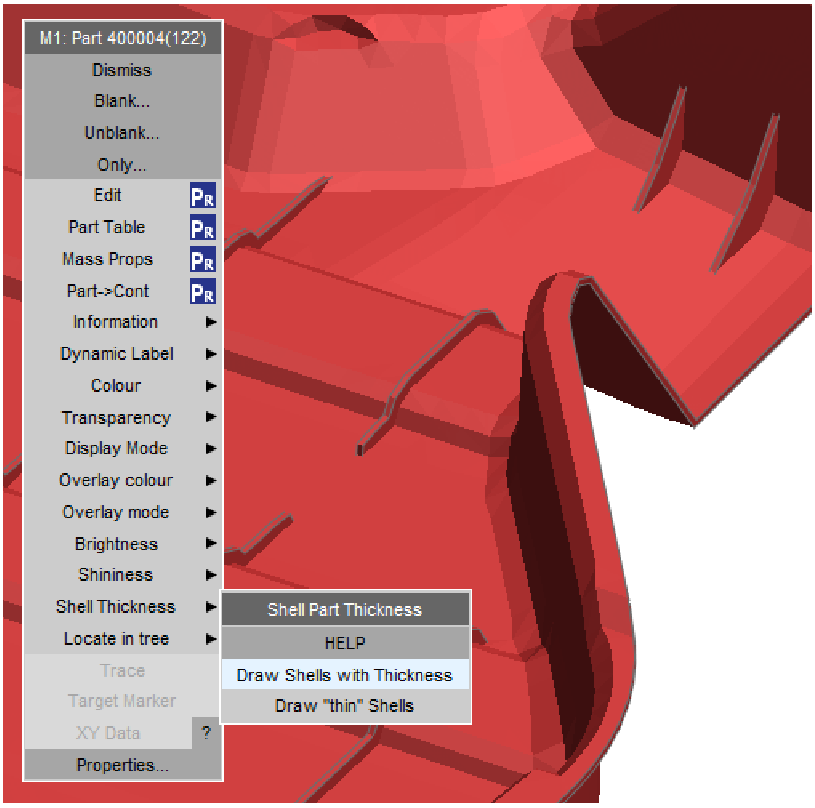
This option can be used to display the data for a specific surface or integration point on all faces of the element.

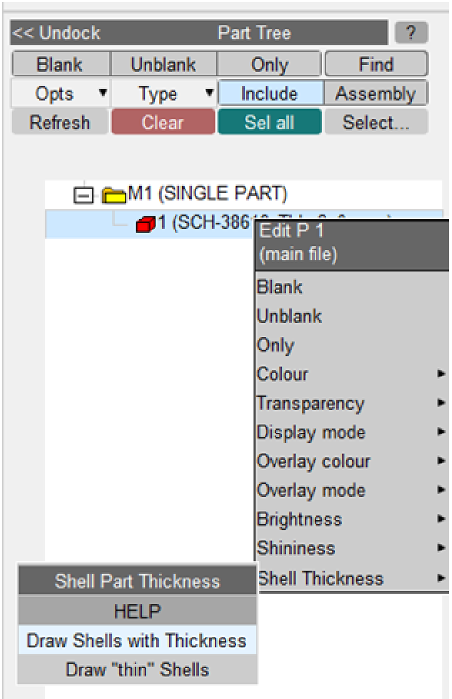


12.22. Drawing Shells with Thickness

Drawing Shells with thickness

The true thickness of 'thin' shells (*ELEMENT_SHELL) can also be visualised in D3PLOT. To preserve graphics speed, shell thickness is controlled on a per-part basis. You can switch on the thickness option for individual parts using either Quick Pick or the Part Tree.





Which thickness value is drawn?

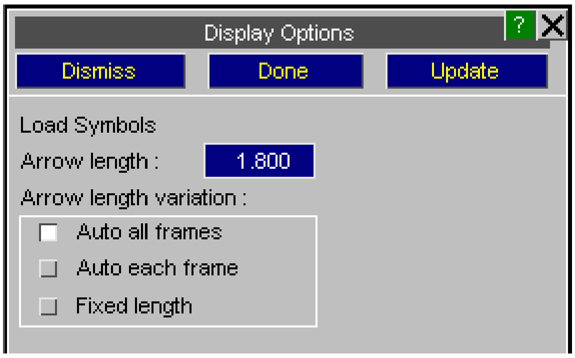
- If ENGFLG set on *DATABASE_EXTENT_BINARY, LS-DYNA will write an **average** shell thickness for each shell (average is relevant in the case where thickness varies between nodes). D3PLOT will then draw shell thicknesses by averaging the values from LS-DYNA at nodes (so an average of averages). So *ELEMENT_SHELL_THICKNESS is considered, but beware averaging.
- If ENGFLG not set, D3PLOT takes values from *SECTION info in ZTF file (so *ELEMENT_SHELL_THICKNESS is ignored in this case).

12.23. Loads

Loads

By default load symbols are drawn based on 4% of the longest model dimension. The symbol size can be adjusted manually in the "Arrow length :" textbox.

Load symbols can be drawn either with a fixed length regardless of the magnitude of the load (Fixed length); variable length normalised to the largest magnitude load of each type in the current state (Auto each frame); or variable length normalised to the largest magnitude load of each type across all states (Auto all frames). By default symbol length is set to "Auto all frames".



12.24. HIDDEN_OPTIONS... Menu: Setting Hidden-Line Display Options

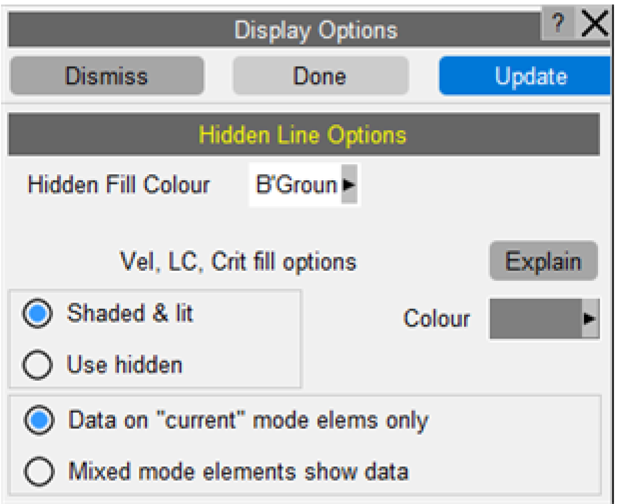
HIDDEN_OPTIONS... menu: Setting hidden-line display options

This menu controls the algorithm and resolution used for creating hidden-line plots when in 2D mode. (It has no influence on 3D mode plots.)

It also controls the element fill colour used for hidden-line plots.

See:

FILL_COLOUR...	To control hidden-line fill colour (all modes)
Vel, LC, Crit fill	Controls appearance of element fill in these plotting modes where data vectors are imposed on top of "structure".



12.24.1. Hidden Fill Colour... Setting the Hidden Line Fill Colour for 2D and 3D Elements

Hidden Fill Colour ... Setting the hidden-line fill colour for 2d and 3d elements

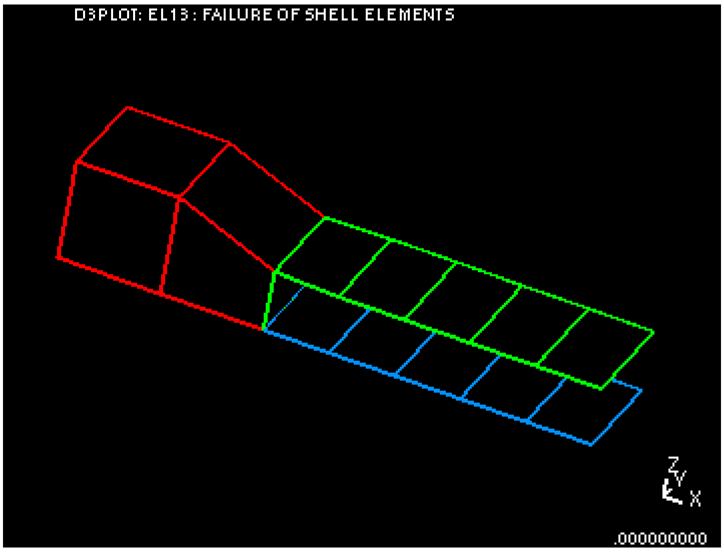
By default 2D and 3D elements are filled with the current background colour, whatever that may be.

However with the popup next to **Hidden Fill Colour** you may choose any other colour. The button shows the current colour (here "B'Ground").

The following three images demonstrate how the fill colour can be modified:

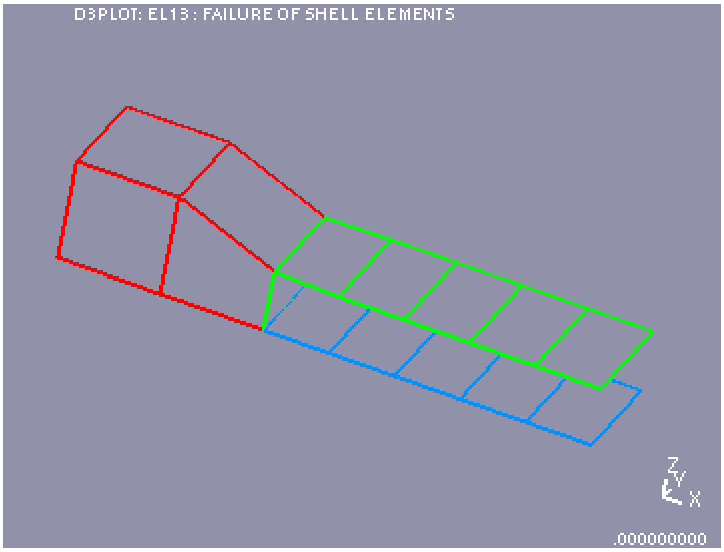
(1) This image shows the default behaviour:

Black background with element infill in the background colour.



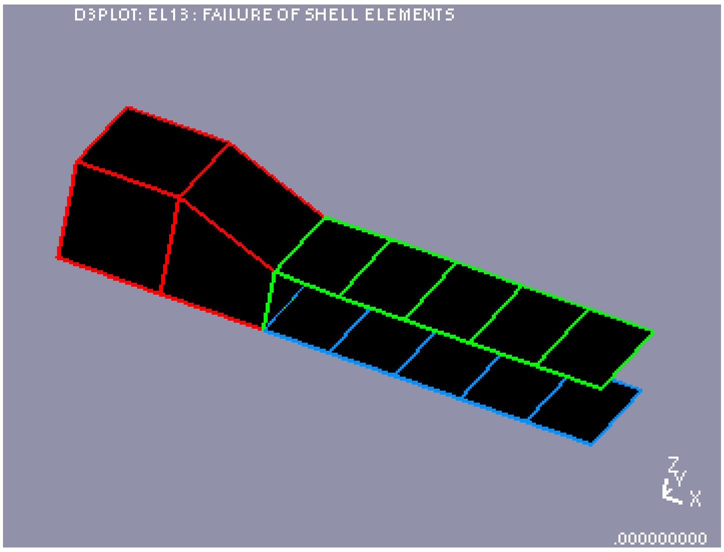
(2) In this image the background colour has been converted to grey.

The default infill is still background colour, so elements are also filled in with grey.



(3) In this image the background is still grey.

The hidden-line infill colour has been set explicitly to black.

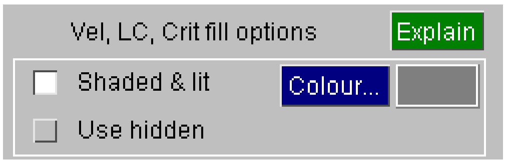


12.24.2. Vel, LC. Crit Fill Options

Vel, LC, Crit fill options

Controlling the appearance of the underlying elements in these "vector data on structure" type plots.

From D3PLOT 9.4 onwards the default appearance of the underlying elements in these plotting modes is shaded and lit, using the default colour of grey. A different colour may be chosen using the **Colour...** popup menu.

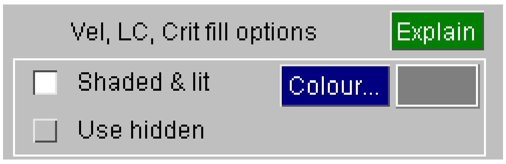


This is a change from pre-D3PLOT 9.4 behaviour in which such elements were drawn in the current "hidden" mode, and it is possible to revert to the previous behaviour by selecting the " **Use hidden** " option instead.

These setting can also be defined via the oa_pref file preferences:

d3plot*vector_fill_mode:	**SHADED** or **HIDDEN**
d3plot*vector_fill_colour:	A standard colour (e.g. **WHITE** , etc) or RGB mixture **0xRRGGBB**

From D3PLOT 9.4 onwards element-derived data vectors will only be drawn on elements which have their display mode set to "current". Those with their mode set to "shaded", "hidden" or "wireframe" will not receive data vectors.



This does not apply to plots of nodally-derived data, e.g. velocity vectors, which will be shown on all unblanked nodes. To limit the display of such data use blanking to prevent the display of nodes at which you don't want to see data.

This too is a change from D3PLOT 9.3 behaviour, and to revert to the previous appearance select " **Mixed mode elements show data** " instead.

This setting can also be defined via the oa_pref file preference:

d3plot*mixed_vector_data:	**SHOWN** or **NOT_SHOWN**

12.25. FREE_EDGES... Menu: Controlling Free Edge Display of Element Borders

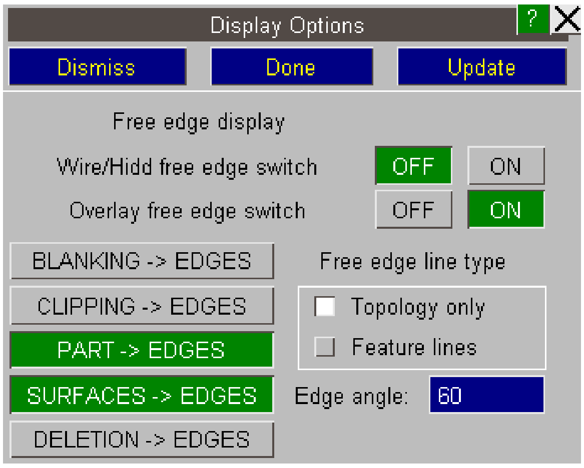
FREE_EDGES... menu: Controlling free edge display of element borders

LI line and **HI** hidden-line plots, and the hidden-line overlay of element borders normally draw all element edges. In "free-edge" mode you can choose to draw only those borders which are "free" edges.

Two separate options are provided for controlling the display of free edged in Line/Hidden plotting and in all other plotting modes. The defaults for these 2 options are:

Line/Hidden Defaults to ***off***

Other overlays Defaults to ***on***



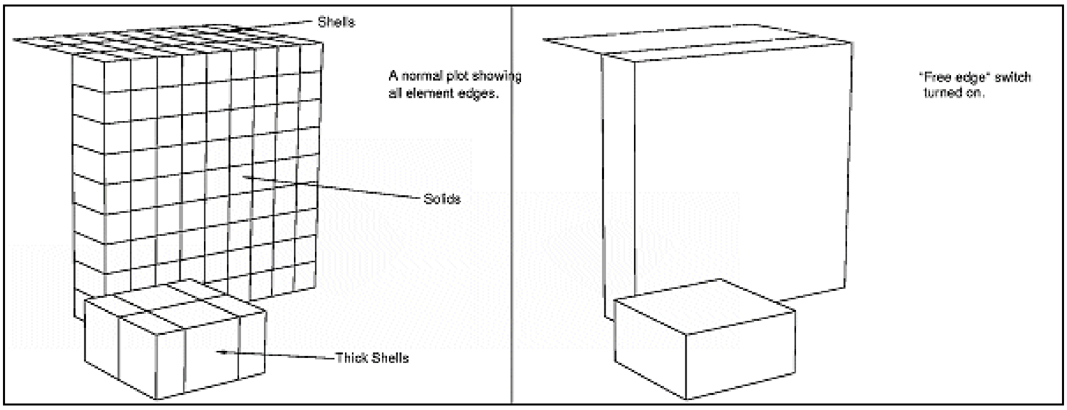
"Free" edges can be displayed either as topological free edges or as "feature" lines: this is explained in more detail below.

Overlays apply to all plotting modes that are not wireframe of hidden, and overlay display itself is controlled in the **OVERLAY** panel. (See [OVERLAY... Controlling the hidden-line overlay of element borders on data plots](#))

Individual element overlay styles can also be set in the **PROPS** panel ([Properties: Controlling Colour, Drawing Style, Transparency, Lighting Attributes and Overlay of Entities](#)).

What is a "free" edge?

A purely topological "free" edge is defined as an element (or face) border that is attached to only one element. This is illustrated in the two figures below.



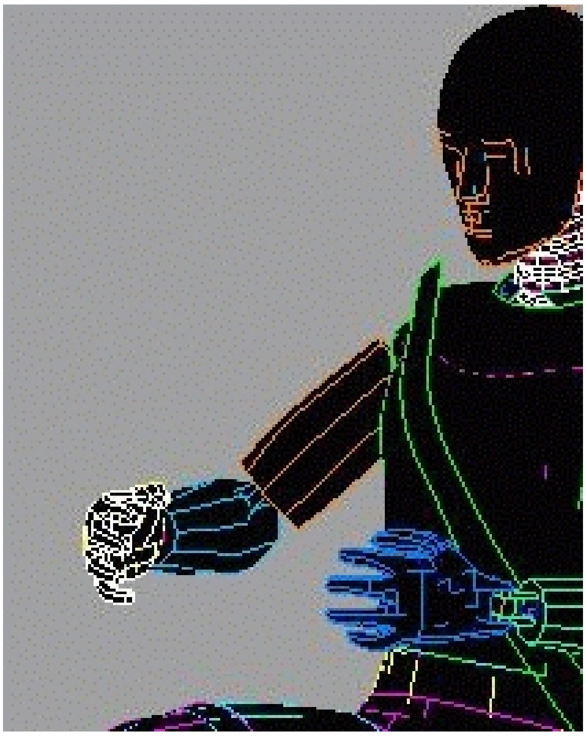
What is a "feature line"?

The purely topological definition of a free edge can sometimes give unsatisfactory images since it is dictated purely by element connectivity, and not by the actual shape of the mesh.

A "feature line" occurs when adjacent elements, or faces of adjacent 3D elements, have outward normal vectors that are more than the "edge angle" apart. This has the effect of inserting extra lines where the mesh changes shape, giving a better idea of the underlying surface.

The "edge angle" is the same as that used to denote sharp edges during smooth shading: see the Edge Angle notes in the section on lighting.

Free edge overlay



Feature line overlay

In this image purely topological free edges have been used. The image is understandable but a lot of detail is absent from areas of mesh of the same part id.

In this image feature lines with an "edge angle" of 20 degrees have been used. This has resulted in lines appearing in previously empty areas: in particular the mouth and eyes have acquired some detail, and the arms are better defined.

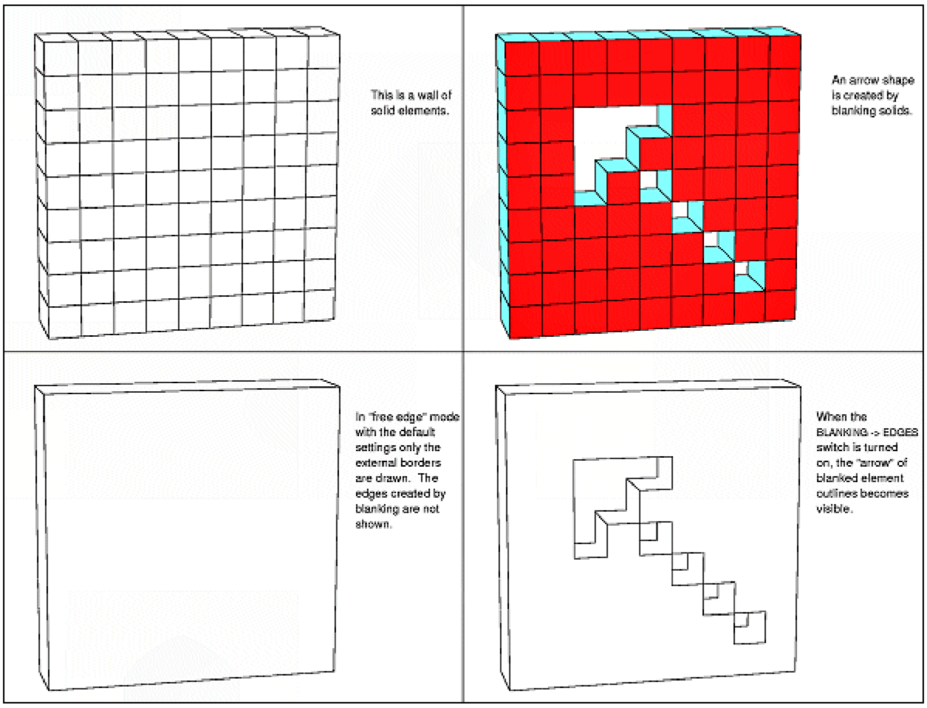
Feature lines are logically ORed with free edges when selected, and they can be used both as overlay on shaded/contoured plots, and as the edging mode for wire/hidden plots. They normally involve more vectors than pure free edges, so they take longer to draw and may have an adverse effect on animation speed.

Modifying the topological definition of a "free" edge

The definition above is open to modification: blanking elements can create free edges, as can volume-clipping and the borders between element materials (or contact surfaces). You can control whether or not each of these categories apply by setting the following switches:

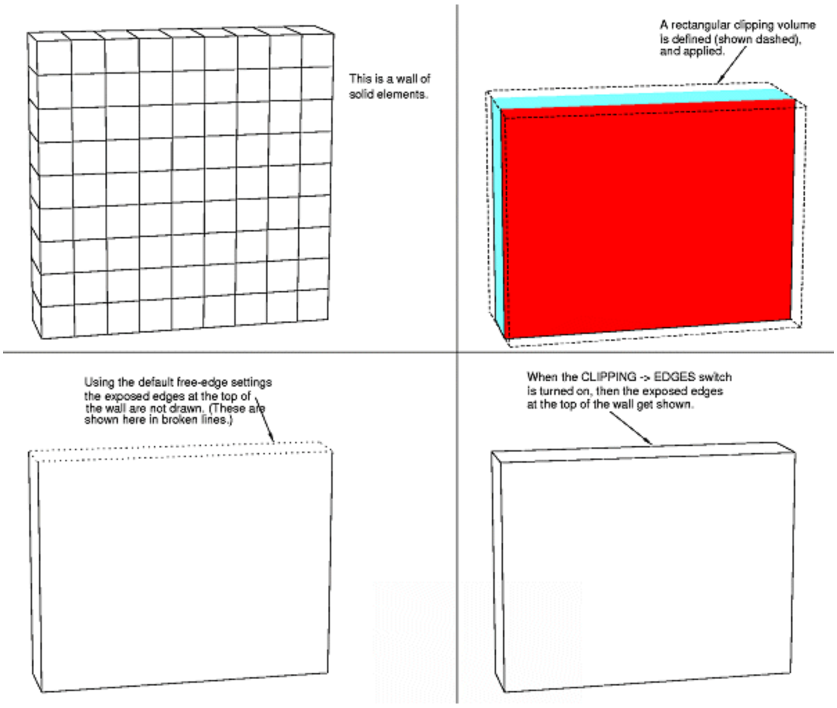
BLANKING -> EDGES Influence of Blanking on free edge display.

Normally blanking elements does not create "free" edges, but if you turn the **BLANKING** -> **EDGES** switch on then the edges created by blanking are shown. This effect is shown in the figure below



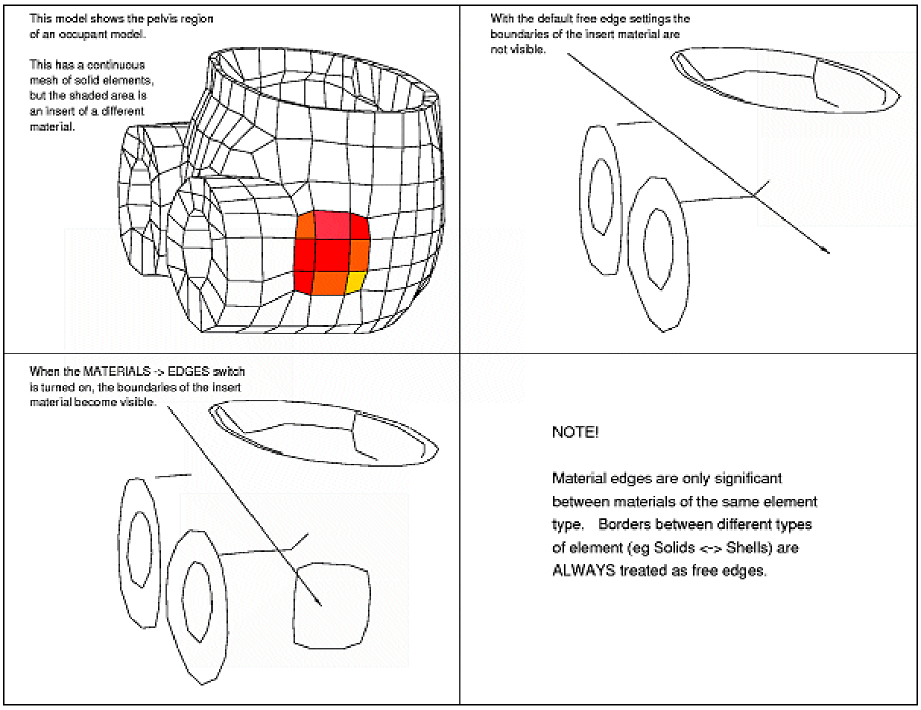
CLIPPING -> EDGES Influence of Volume-Clipping on free edge display

In the same way that element borders created by blanking do not, by default, create free edges; those exposed by volume clipping do not either. This is illustrated in the figure below where a clipping volume is used to remove solid elements from the same wall as in the previous example.



MATERIALS -> EDGES Influence of material boundaries on free edge display

The border between materials of the same element type does not, by default, qualify as a free edge. (Borders between different element types always qualify as edges.) If **MATERIALS -> EDGES** is turned on then such borders are treated as free edges. This is illustrated in the figure below.



SURFACES -> EDGES Influence of contact surface boundaries on free edge display

You can think of contact surface boundaries in the same way as material boundaries: the different surface ids equate to the different material ids, and the same edge definition logic applies.

However:	The default setting of the **SURFACES -> EDGES** switch is on. The reason for this is that it would be very unusual to have two adjacent contact surfaces forming a single (geometric) surface, but it is quite common to have two or more coincident surfaces. In the coincident case it is useful to see the edges.

Notes on **FREE_FACE** options:

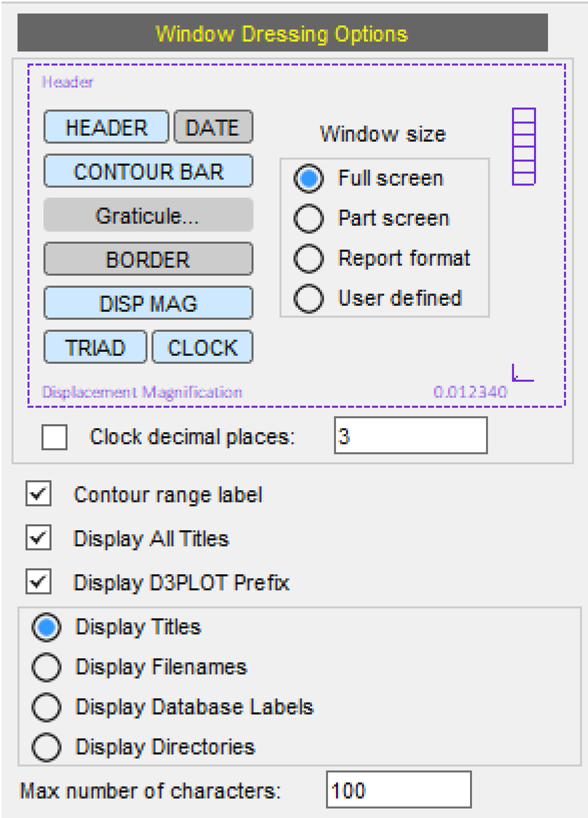
- Free edges are only computed for 2 and 3-D entities, that is: solids, thin shells, thick shells and contact surface facets.
- Free edge computation is worked out separately for each element type. For example an edge common to a shell and a solid will still be treated as a free edge.
- Where elements are deleted, due to material failure, free edges may be created at that and subsequent complete states.
- A "line" mode plot with free edges only is the fastest way of drawing something in D3PLOT since it combines minimal computation with only a small amount of screen vectors. It is a good way of assembling a quick animation
- Precomputed free-edge dynamic viewing is available: see Graphics modes during dynamic viewing .

12.26. WINDOW_DRESSING... Menu: Controlling Screen Appearance

WINDOW DRESSING... menu: Controlling screen appearance

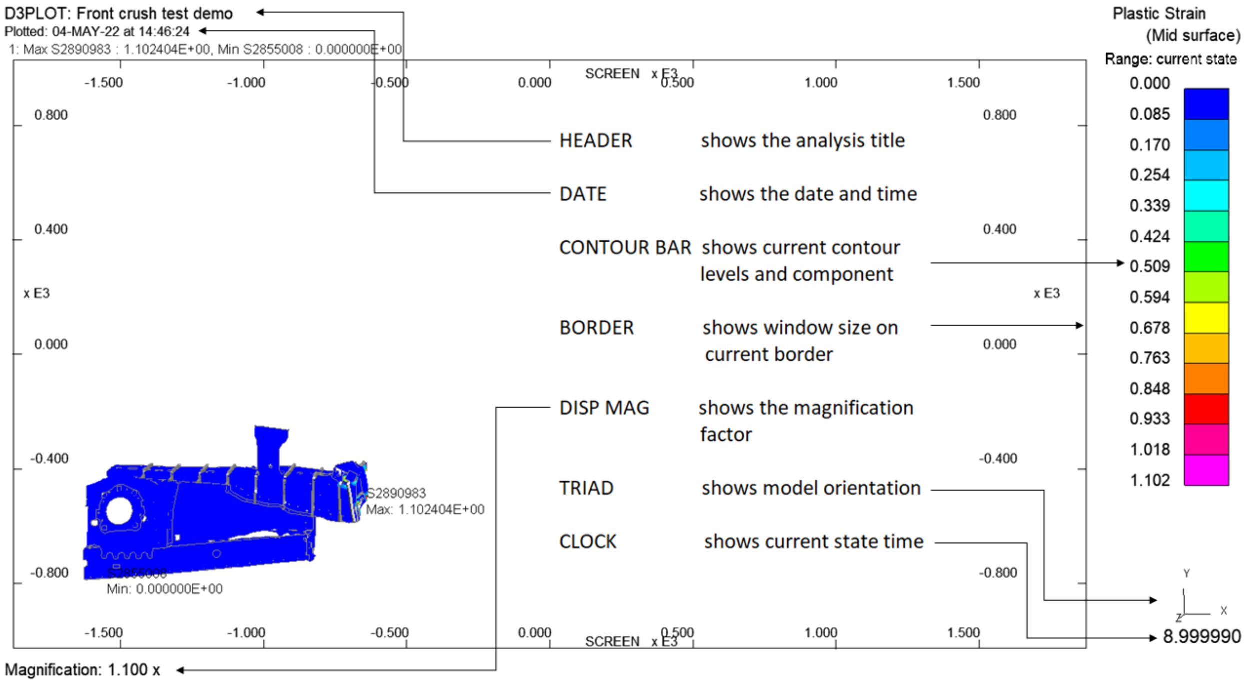
The "dressing" that can be added to plots: date, header, triad, etc is controlled from this window.

It also controls the "soft" window size.



Controlling "Window Dressing": HEADER , DATE , BORDER , DISP MAG, CLOCK, TRIAD , CONTOUR_BAR

This figure shows the main "Window Dressing" attributes. Each of these can be switched on or off at will, the default being everything except the **DATE, GRATICULE** and **BORDER** are on. The **Window Size** shown in this example is **User Defined** so that the borders are visible, but the default is **Full Screen** .

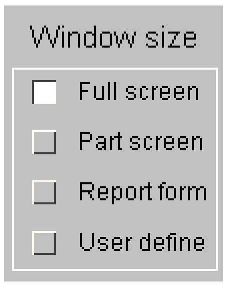


These options are all straightforward, with the exception of the **GRATICULE** (see Graticule for more details).

Controlling the "soft" window size: W indow Size

By default the graphics can extend over the complete window area, and no "soft" window clipping is in force. You can restrict the graphics to a sub-set of the screen, i.e. to a "soft" window, by settings a **Window Size** .

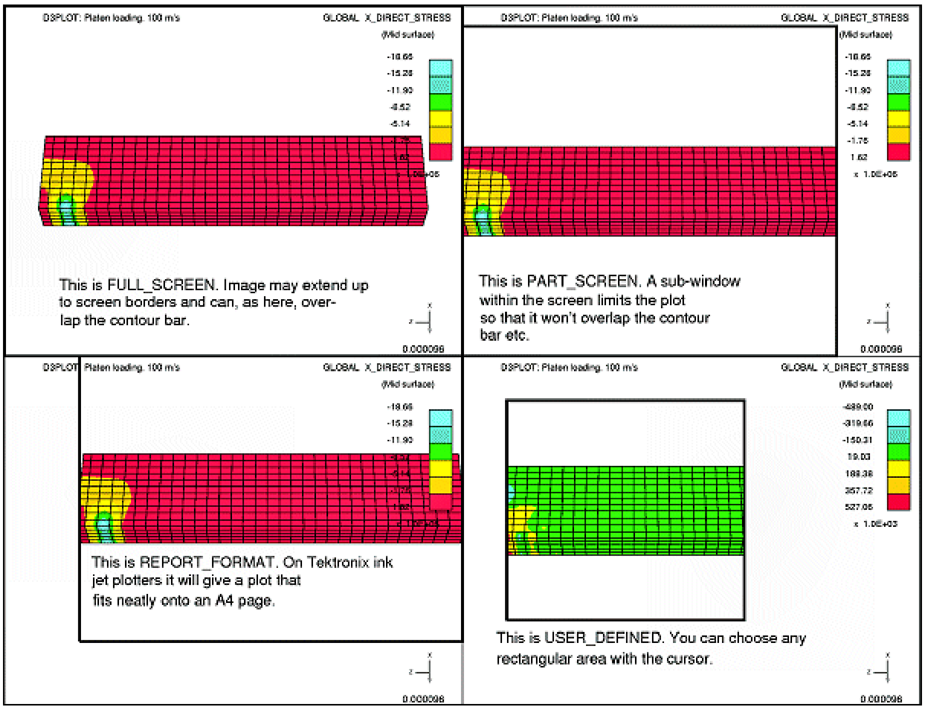
This is controlled by the radio buttons in the middle of the **Window Dressing** control panel, as shown here.



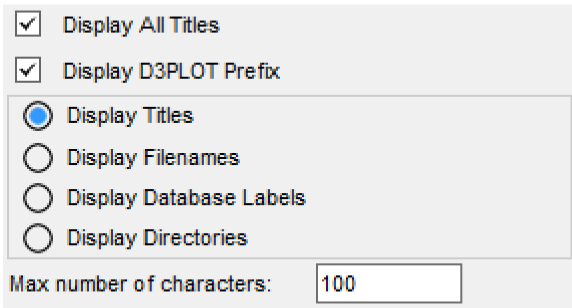
There are four pre-programmed options:

Full screen	(Default) use the whole screen
Part screen	Leave contour bar and header clear
Report format	Set dimensions for local hard-copier
User defined	Set screen rectangle using the mouse

The meaning of each of these is shown in the figure below:



If a window contains multiple models then by default D3PLOT will display multiple titles in the window. Turning off the **Display All Titles** option will make D3PLOT display just the title form the 1st model in the Window.



Contour Range Label	(Default) The option will display the range label over plots for each window.
Display Titles	(Default) The option will display the title for each model.
Display D3PLOT Prefix	(Default) The option will display the "D3PLOT" prefix for each model.
Display Filenames	Instead of displaying the title of each model this option will display the filename of each model.

Display Database Labels	If a model had been read in using the model database option (see Select Models From Database) then instead of displaying the title of each model this option will display the label used to identify the model in the database.
Display Directories	Instead of displaying the title of each model this option will display the parent directory and filename of each model
Max number of characters	This controls the maximum number of characters displayed in the header.

12.27. Graticule

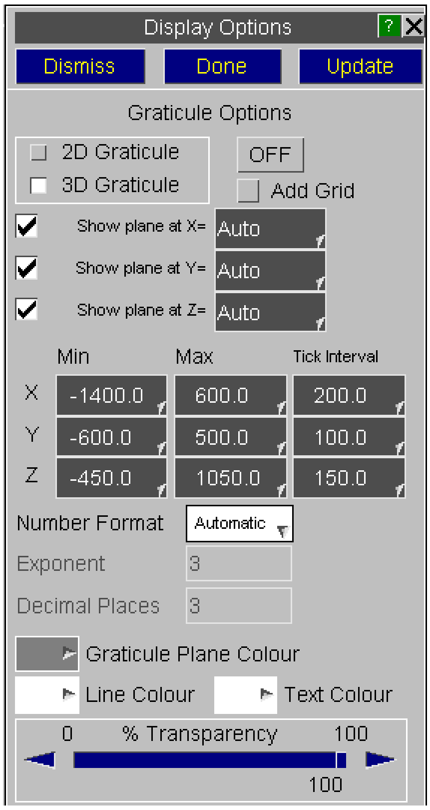
Graticule

This can be used to display the current model dimensions.

The graticule can be drawn in either 2D or 3D.

The format of the numbers on the graticule can be set automatically by D3PLOT or you can manually select the number of decimal places and the exponent value to display.

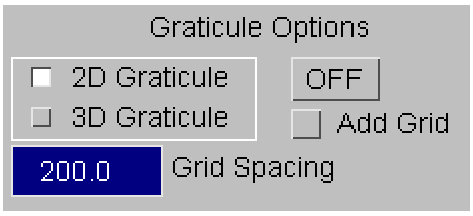
The line and text colours can be modified if necessary.



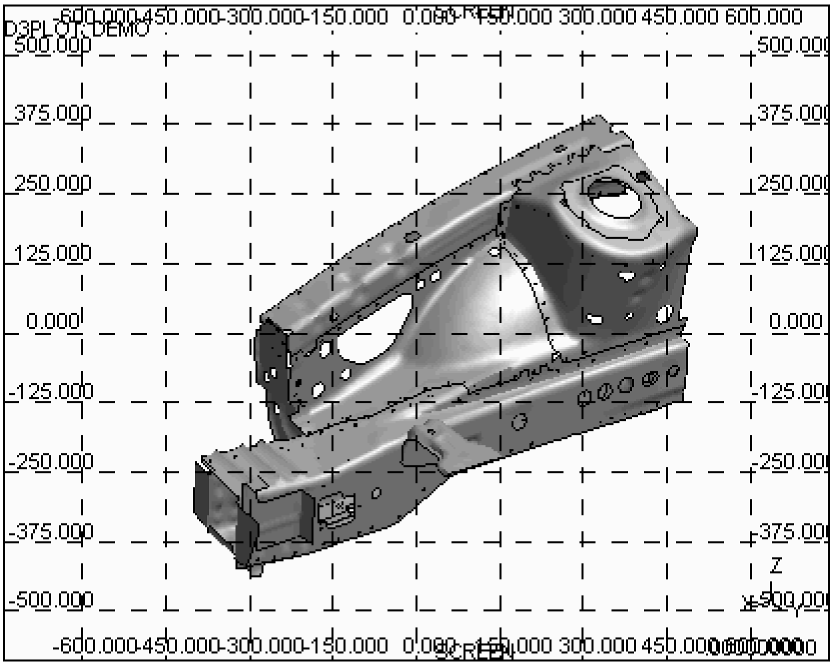
12.27.1. 2D Graticule

2D Graticule

If the grid spacing can is set to 'Auto' D3PLOT will calculate a sensible value. If you want you can input a value manually.



With the 2D graticule the space system used to display the model dimension depends on the current view.



Model space Is used if the view is orthogonal down one of the screen X, Y or Z axes. The appropriate XY, YZ or XZ coordinates are shown, and these move as the model moves (try dynamic translation and you'll see).

Screen space Is used if the view is not orthogonal. This just shows the current window dimensions (X = 0 - 4095, Y = 0 - 3129). It is only useful for setting up volume clipping using screen space orientation.

If a **GRID** is added it draws a grid on the screen at the current tick mark interval.

12.27.2. 3D Graticule

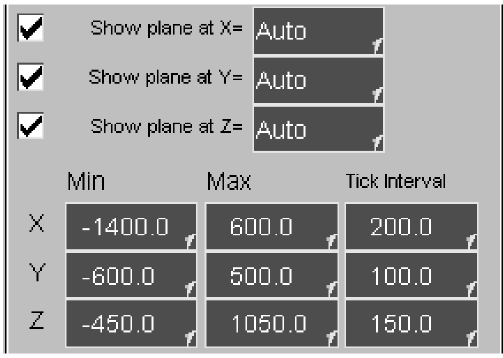
3D Graticule

The 3D graticule option will produce 3 planes aligned with the global x, y and z axis which show the model bounding box.

The display of each of the 3 plane can be turned on and off separately as required.

As well as specifying the minimum and maximum dimensions for each plane the location of each plane can also be specified along with the grid interval.

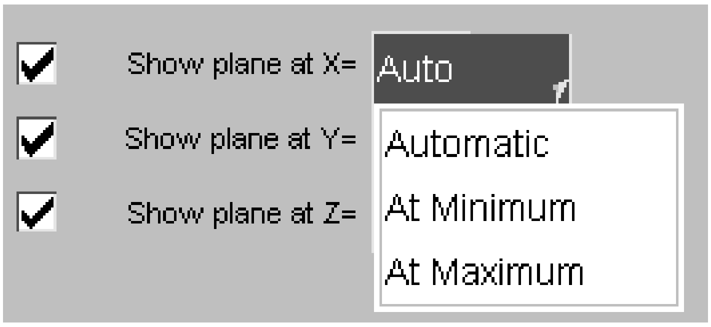
By default D3PLOT will automatically calculate all the graticule plane values. If the user modifies any of the values then the text box colours will change to WHITE text on a DARK BLUE.



By default D3PLOT will automatically calculate the location of the 3 graticule planes based on the model dimension. The location of each plane can be changed by entering the new location in the text box.

Alternatively 3 pre-set locations can be selected.

Automatic	This is the default option. D3PLOT will automatically locate the plane at either the minimum or maximum value so that it is positioned behind the model from the users view point. As the model is rotated D3PLOT will adjust the plane location as required.
At Minimum	The plane will automatically be located at the minimum value for the axis. If the axis minimum is modified by the user the plane location will automatically update.
At Maximum	The plane will automatically be located at the maximum value for the axis. If the axis maximum is modified by the user the plane location will automatically update.

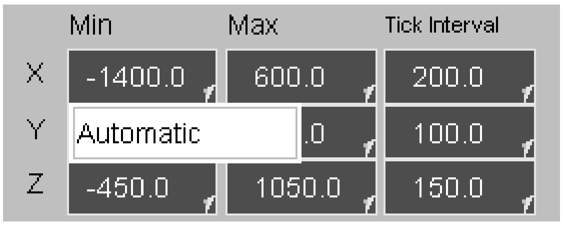


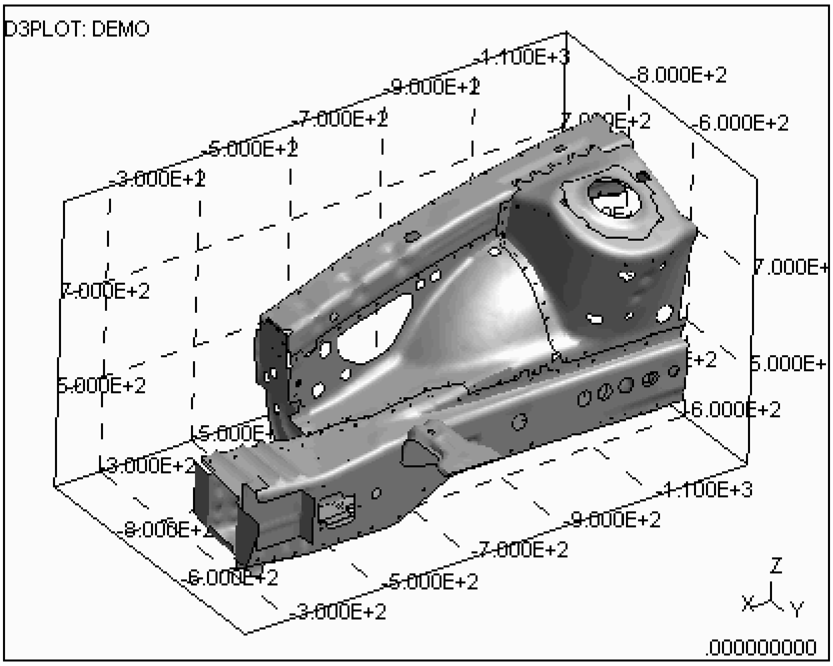
By default D3PLOT will automatically calculate the minimum and maximum values used to display each plane along with the interval between the values displayed.

The minimum and maximum values along with the tick interval can be changed using the text boxes. If any of the values are changed then the text box colours will change to WHITE text on a DARK BLUE.

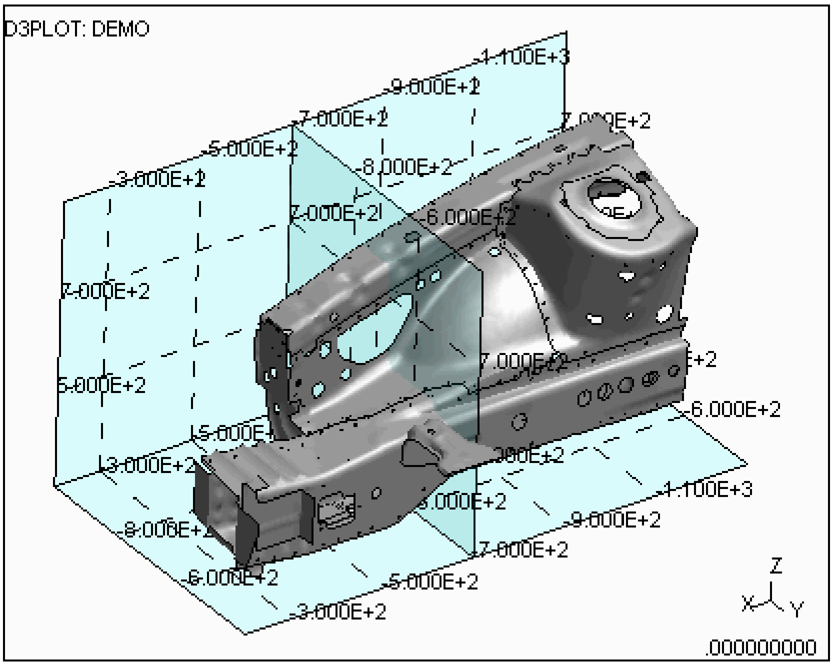
All of the values can be reset to **Automatic** using the popup menu.

If the Tick Interval is set to **Automatic** D3PLOT will adjust the tick spacing if required as you zoom in and out.





If necessary a transparency value and colour can also be set for the 3 planes



12.28. Fonts

Fonts

Font types and sizes can be set for various text displays.

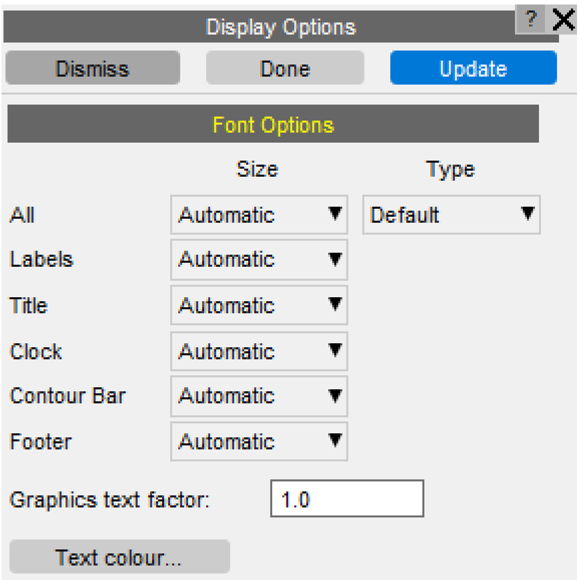
The font size can be set individually for:

- **Labels**
- **Title**
- **Clock**
- **Contour Bar**
- **Footer**

Setting the size to Automatic, D3PLOT will select a font size which will vary with window size. If an explicit point size is selected the text will stay constant no matter the size of the window.

The font type is applied to all graphics text.

The Graphics text factor scales the font size if it is set to Automatic. It will have no effect if an explicit point size has been selected.



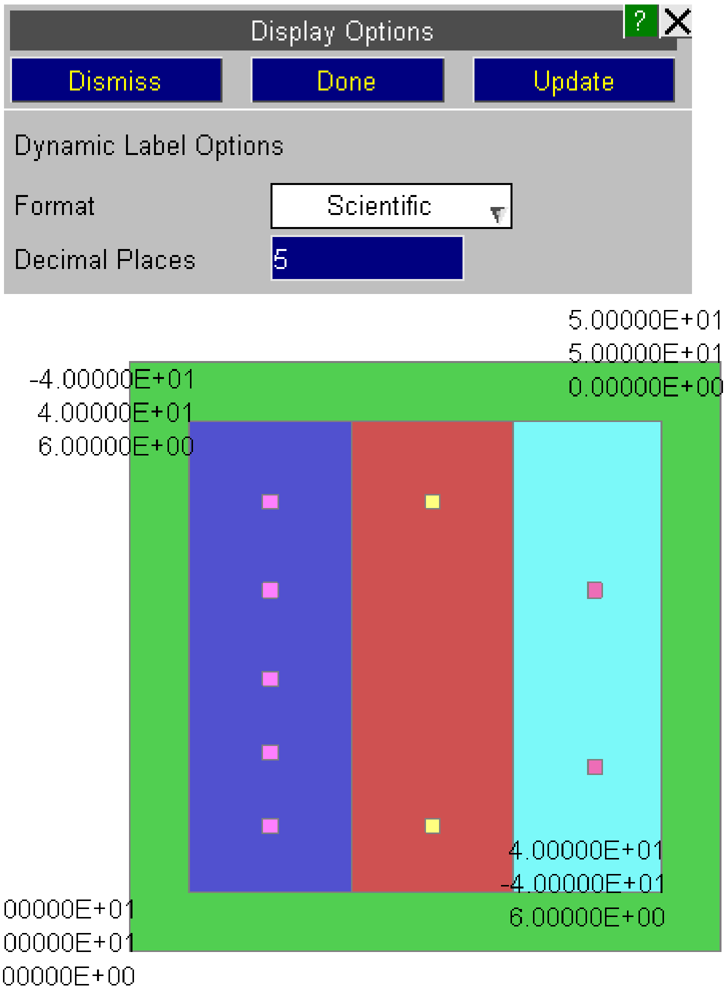
12.29. Dynamic Label Format

Dynamic Label Format

One of the following numeric formats can be specified for dynamic label values and coordinates:

Automatic
Scientific
General

For **Scientific** and **General** formats, the number of decimal places may also be specified.



12.30. Material Attribues

Material Attribues

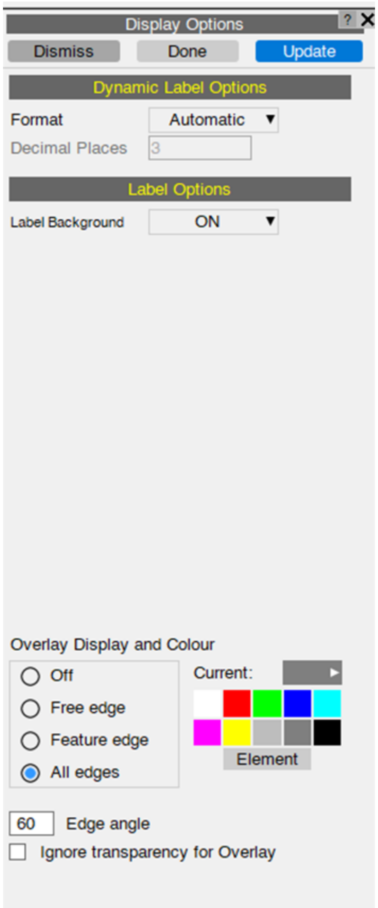
Information on the Material Attributes panel can be found in Material Attributes

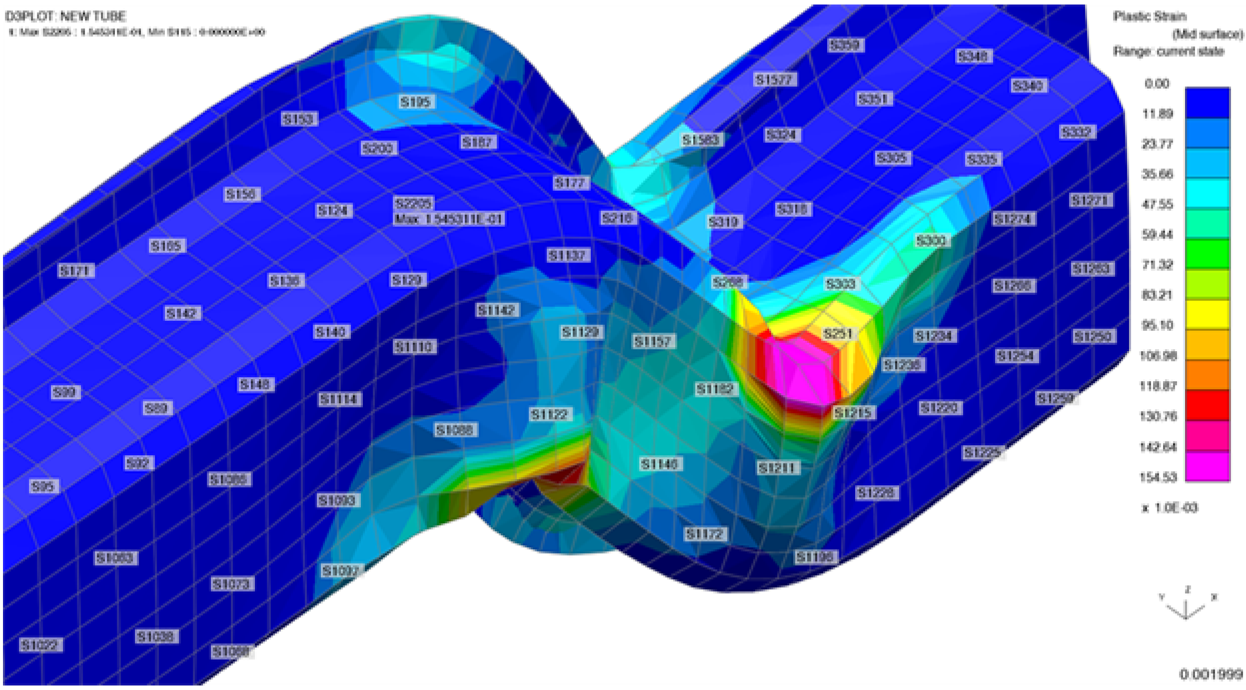
12.31. Label Background

Label Background

Labels now have a background to increase the legibility of the text.

The background can be turned off in Display Options → Label Options → Label Background or by setting the preference d3plot*label_background.



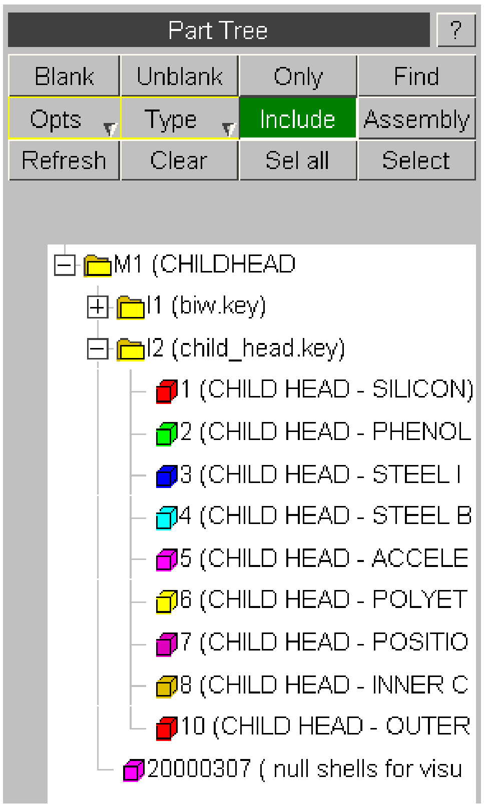

D3PLOT: NEW TUBE
Plastic Strain
(Mid surface)
Range: current state
0.00
11.89
23.77
35.66
47.55
59.44
71.32
83.21
95.10
106.98
118.87
130.76
142.64
154.53
x 1.0E-03
0.001999

13. Part Tree

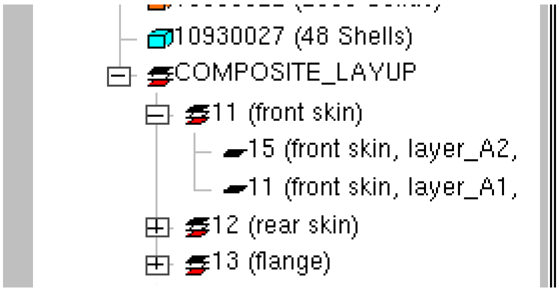
PART TREE

This enables quick navigation around a Model and helps with blanking etc. The Part Tree is available from the tab:

The part tree defaults to a view of the parts within the mode. If a `.ztf` file is present, the model hierarchy by INCLUDE file will be displayed



If the model includes composite plys and a `.ztf` file is present, the composite plys are listed at the end of the part tree. The plys are grouped by LAYUP if layups have been set-up in PRIMER. Under each layup, the plys are ordered by their position in the layup.



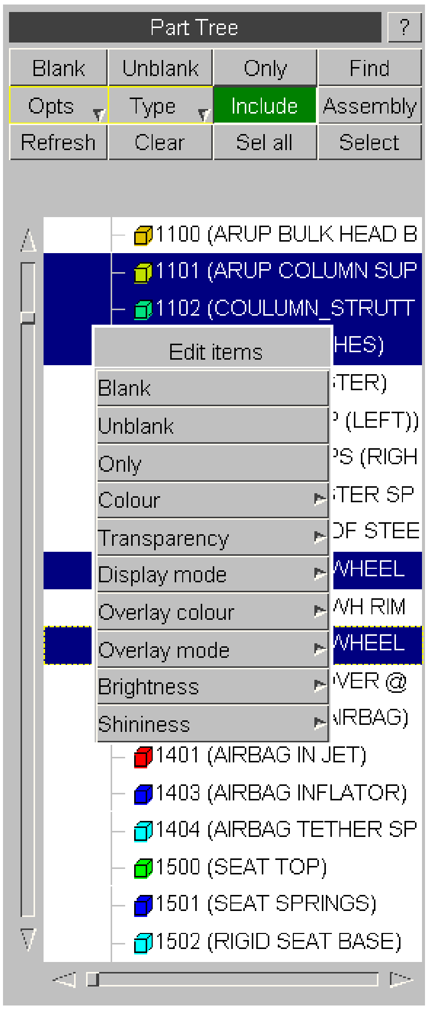
13.1. Part Tree Behaviour

Part Tree Behaviour

Items can be selected by left-clicking anywhere on their row. Where selecting more than 1 item would be valid you can hold <ctrl> whilst clicking to select multiple items. Alternatively the <click> (start of range) .. <shift><click> (end of range) method (cf Windows) may be used.

Clicking on the [-] button next to models / include files / assemblies will collapse branches. Collapsed branches will have a [+] button which when clicked will expand the branch.

Right-clicking on an item or a selection of items produces a pop-up menu with the options shown on the right (not all of these options will be available for some selections).

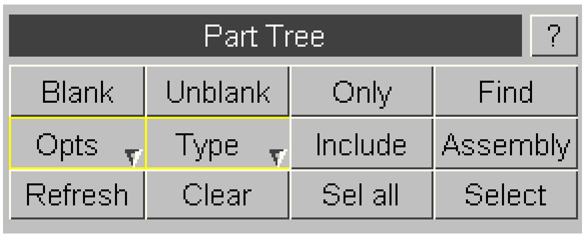


Blank	Blanks the item
Unblank	Unblanks the item
Only	Blanks all other items and unblanks the item
Colour	Colours the items (or elements associated with the item)as selected
Transparency	Sets the transparency the items (or elements associated with the item)as selected
Display Mode	Marks the item to be drawn with a particular method (wireframe, hidden, shaded or current)
Overlay Colour	Sets the item to be drawn with a particular colour overlay
Overlay Mode	Sets the style of Overlay the item is to be drawn with (see OVERLAY... Controlling the hidden-line overlay of element borders on data plots)
Brightness	Controls how light or dark the colour of the item is when illuminated, but the effect is to add matt colour (not whiteness, which would make it look shiny).
Shininess	Adds white highlights, but no colour, to make the object look shiny.

13.2. Part Tree Top Menu Bar

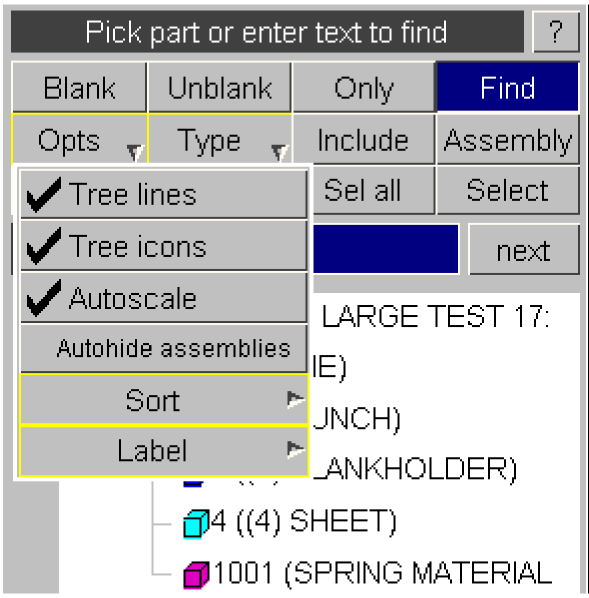
Part tree top menu bar

The top menu bar allows quick access top common functions, as well as controlling how the part tree behaves and is displayed.



Opts

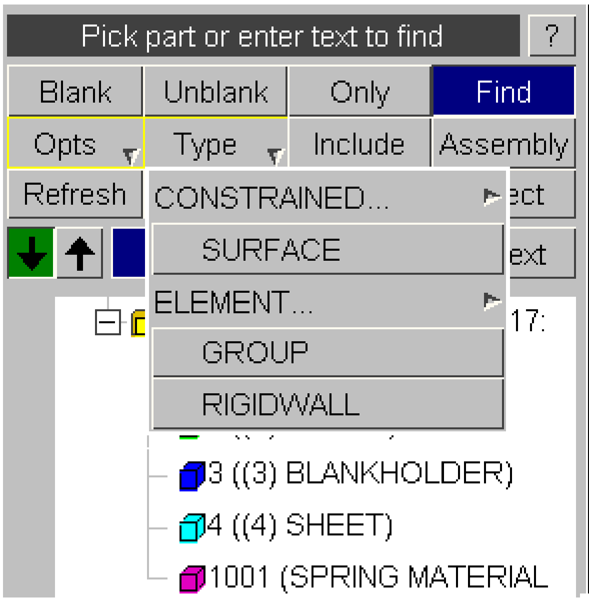
There is a range of options for controlling the part tree available via the **Opts** pop-up menu. These include how items are labelled and ordered as well as whether assemblies, tree lines and icons are drawn.



Type

From the Type pop-up menu it possible to select a variety of different item types to be displayed in the tree in addition to parts. These appear below the parts in the tree, and most of the options (edit, blank etc) are available through the "right-click" menu.

The GROUP option available here can also be used to list the HBM visualisation entities related groups in the Part tree. These groups can be read from a ***.vis** file from the "Groups >> Read" panel. The detailed information about this is available here.



Blank / Unblank / Only

One use of the part tree is as easy way access to blanking commands. **Blank** , **Unblank** and **Only** (blank all other items) and commands can be applied to the currently selected items.

Sel all / Clear

The **Sel all** and **Clear** buttons can be used to select all items and empty the selection respectively.

Select

The **Select** button invokes an object menu for selecting parts. Selection can also be made via the Quick Pick option "Locate in Tree". Operations such as blank,colour, etc may then be carried out on the selected entities.

Include/Assembly

The Include and Assembly buttons determine what type of hierarchy is displayed.

These have effect only when the requisite information is available to D3PLOT via the ztf file, and in the case of assemblies, via an assembly file written out by PRIMER.

Assemblies are user-defined hierarchical groupings of parts, created in PRIMER.

Find

The **Find** button gives a search option. Text or an ID number is entered in the text field. D3PLOT finds a part whose title contains the text, or a part with an ID matching the number. The arrows determine whether the search direction is up or down from the current selection. **Next** will find the next matching item. The search will only find matches for currently enabled options (i.e. if id is disabled and items are labelled by name only a search for part 15 will return no matches regardless of its presence in the tree).

14. Search (Quick Find)

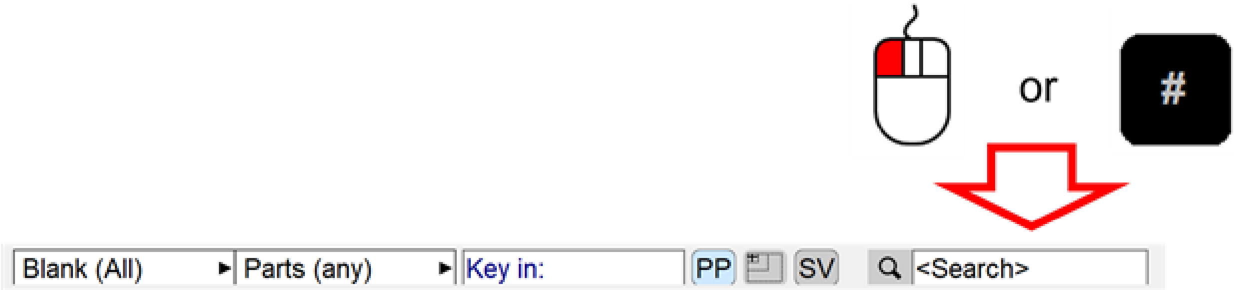
14.1. Introduction

Introduction

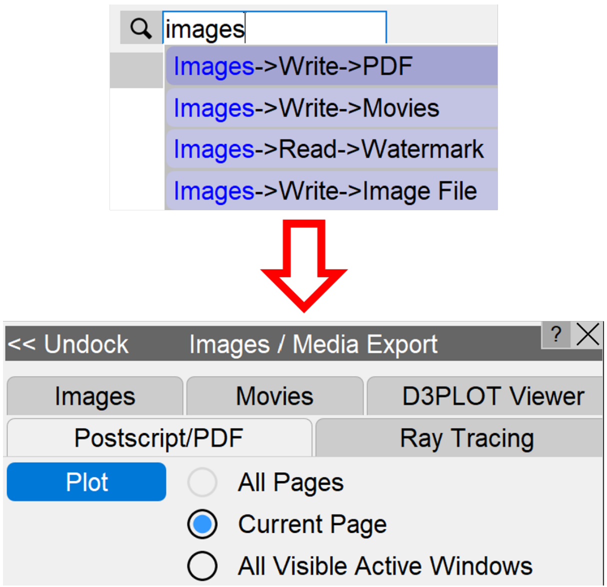
Quick Find can be used to search for and quickly:

- Go to menus / functionality
- Open tutorials

It can be accessed by clicking in the Search box in the top bar or by pressing the '#' key.



Typing in the textbox brings up a list of found items that match the entered text. Items in the list can be selected by clicking on them or by using the up and down arrow keys and pressing enter. The selected item will then perform the task, e.g. open a menu.

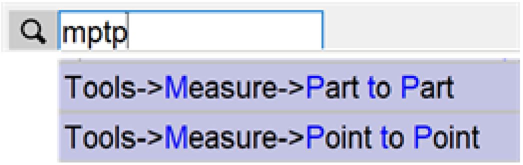


14.2. Fuzzy Matching

Fuzzy Matching

A 'fuzzy' matching method is used to match the entered text with the searchable items. It judges that something has matched when the characters of the entered text appear in the same order as the item that can be searched for.

For example if you type 'mptp' then 'Tools-> **M** easure-> **P** art **T** o **P** art' would be a match, but 'Tools-> **M** easure-> **P** oin **t** Angle' wouldn't because the final 'p' doesn't match. (Note that the search is case insensitive).



Additionally, if the entered search pattern contains spaces and the characters do not all match in the same order then D3PLOT will look to see if the words can be swapped to find a match.

For example 'back image' would find 'Image->Background' even though the words do not appear in that order.

This hopefully makes it easier to find items as you do not need to know the precise search term.

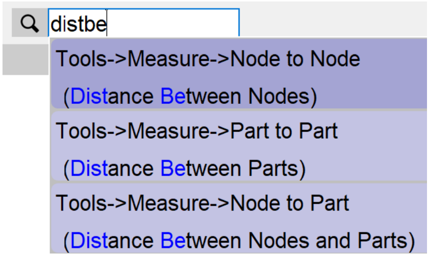
The found items are listed in order of how closely they match the entered text so items that more closely match appear nearer the top of the list. It determines this by assigning a score to each match, with higher scores given to items that contain consecutively matched characters and if the characters appear at the start of words.

14.3. Search Terms

Search Terms

The default search term associated with a menu item is the trail of menus/buttons you would need to manually open/press, e.g. to get to measure part to part you would need to go to Tools, then Measure then Part to Part, hence the search term 'Tools->Measure->Part to Part'.

In addition, some menus have alternative search terms associated with them. For example Measure Part to Part can also be found from the alternative text 'Distance Between Parts':



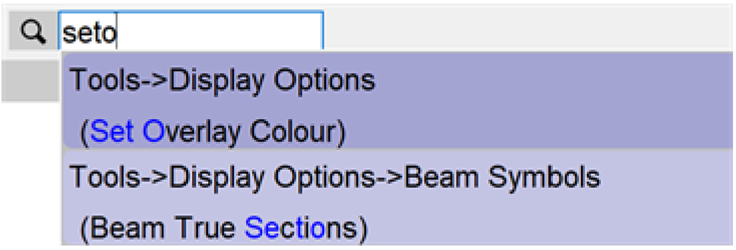
This can be useful for cases where you don't know or can't remember under which menu some functionality lives.

Note that the alternative text appears in brackets under the default search term so you can see how you would get to the menu manually.

If you can't find menus that you know exist in D3PLOT it is likely that you are using different terminology to what we expect. If so, please contact Oasys Ltd and we can add alternative text based on what you are entering as your search text.

Alternative text associated with a menu may also describe some of the features on a menu. For example the overlay colour of elements is set in the Display Options menus, but if you didn't know this it would be hard to find.

In this case the alternative text 'Set Overlay Colour' is associated with this menu:

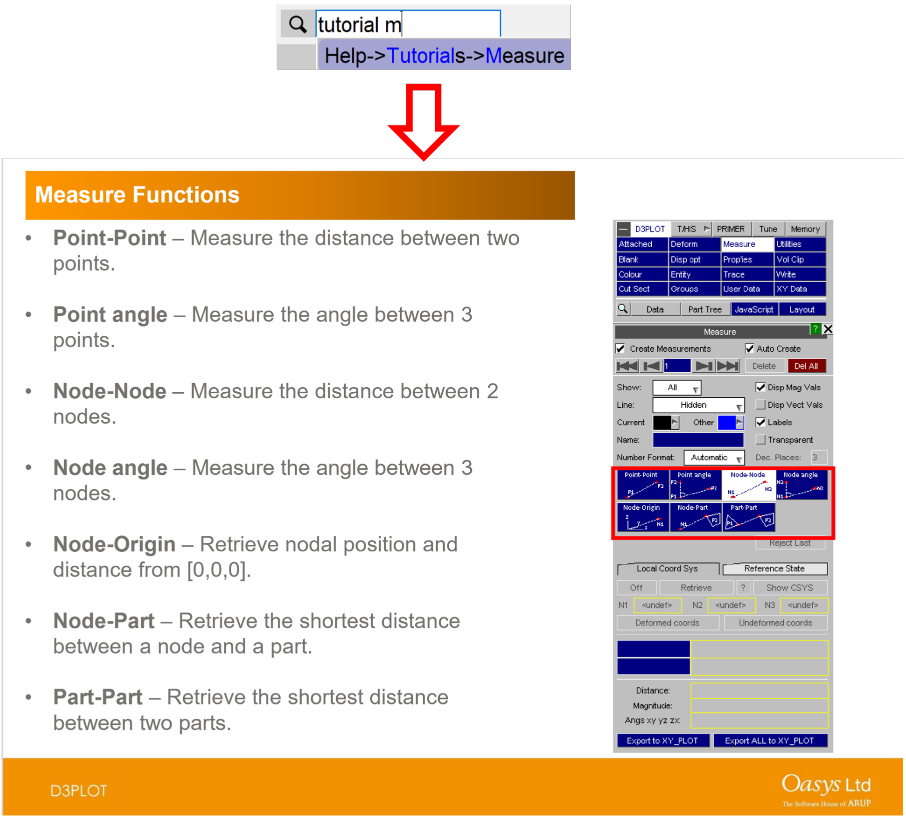


As you can see the alternative text 'Beam True Sections' is also associated with this menu as the switch to select this option is also on the Display Options menu.

14.4. Tutorials

Tutorials

The full installation of the Oasys Ltd software contains some pdf tutorials for various features within the software. They are installed in the $OA_INSTALL/manuals/tutorials/d3plot directory and can be found and opened using Quick Find.

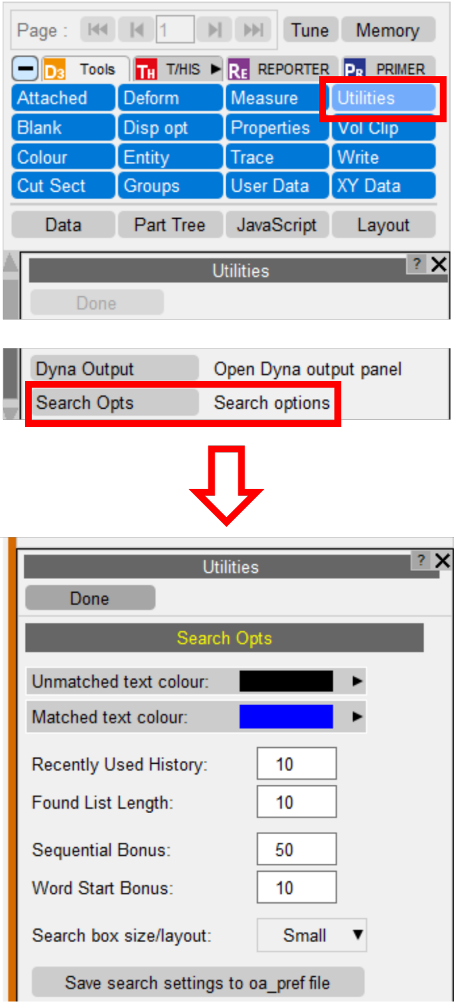


14.5. Options

Options

There are a few options that can be set to alter how Quick Find works. These can be accessed by clicking the Utilities->Search Opts button.

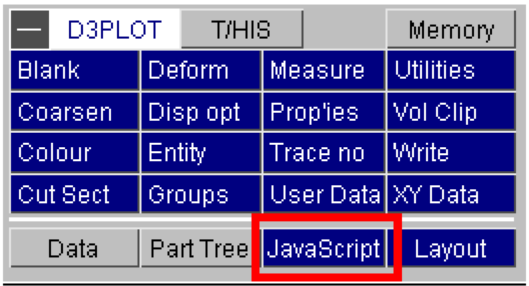
- Save settings to the oa_pref file
- Set the text colours for matched and unmatched characters
- Recently selected items are saved by D3PLOT and appear higher in the list of available options. By default the last ten selected items are saved, but this can be changed here. To turn it off set it to zero.
- Set the maximum number of found items to display in the list
- Specify the Search box size


Page :
1
Tune
Memory
Tools
T/HIS
REPORTER
PRIMER
Attached
Deform
Measure
Utilities
Blank
Disp opt
Properties
Vol Clip
Colour
Entity
Trace
Write
Cut Sect
Groups
User Data
XY Data
Data
Part Tree
JavaScript
Layout
Utilities
Done
Dyna Output
Open Dyna output panel
Search Opts
Search options
Utilities
Done
Search Opts
Unmatched text colour:
Matched text colour:
Recently Used History:
10
Found List Length:
10
Sequential Bonus:
50
Word Start Bonus:
10
Search box size/layout:
Small
Save search settings to oa_pref file

15. The JavaScript Interface

The JavaScript Interface

Programming D3PLOT externally for both batch and interactive use.



15.1. Introduction

Introduction

JavaScript is a freely available scripting language that is normally found performing the "work" behind interactive web pages, however its syntax and structure also make it an excellent tool for providing an externally programmable interface to programmes in general.

Within D3PLOT it is implemented as follows:

- There is a D3PLOT Application Programming Interface (API) within the JavaScript API Reference Manual which provides a range of functions that allow you to interrogate the database, open windows, generate plots, and so on. This is written in a very simple and non-intimidating way, with relatively few functions, that should be easy for non-programmers to use.
- There is also a function which issues "command line" instructions to D3PLOT, making it possible to use the code's full repertoire of command-line commands, meaning that virtually every function in D3PLOT is callable from within a JavaScript.
- There is a special class of "user defined binary (UBIN) data components" that can be created from within a JavaScript, making it possible to generate an unlimited number of new data components which then become available for processing in exactly the same way as the standard ones found in an LS-DYNA database.

Anyone familiar with C or shell script programming will find existing JavaScripts are instantly readable, and can be given minor edits without further ado. For those who are more ambitious a good guide to the language is " **JavaScript, A definitive Guide** " by David Flanagan, published by O'Reilly, ISBN 0596101996.

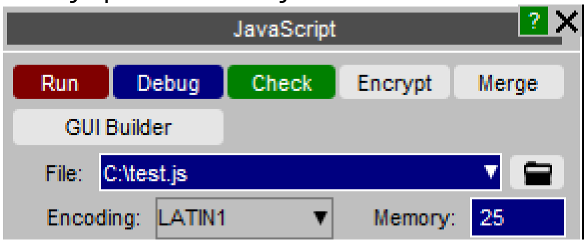
In D3PLOT 17.0 and earlier the implementation supported ECMAScript 5 features of JavaScript. In D3PLOT 18.0 the implementation has been upgraded to support ECMAScript 6 (and newer) features of JavaScript.

The sections below describe how to run JavaScripts in D3PLOT, and summarise the JavaScript API. For details of the API and its functions, and also some examples, see the JavaScript API Reference Manual

15.2. Using JavaScript in D3PLOT

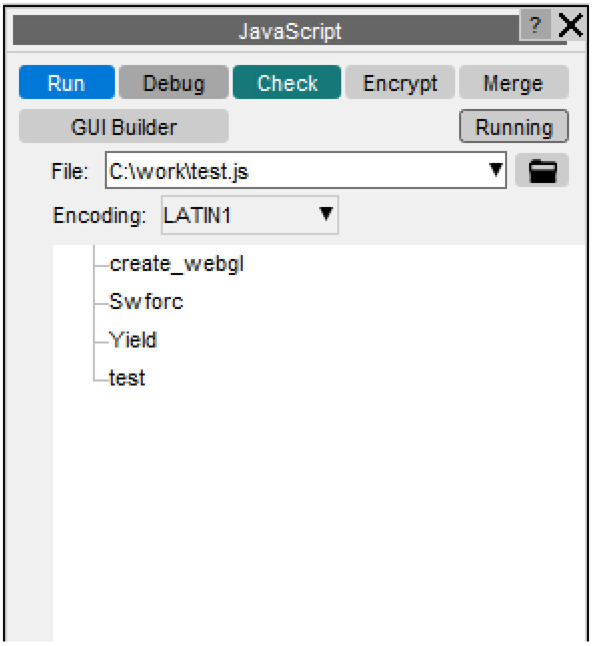
Using JavaScript in D3PLOT

Human-readable Javscripts need to be ***compiled*** , meaning turned from something human-readable into a set of instructions that a computer can understand; and then ***run*** in their compiled form. They can be changed and rerun in their modified form at any time without having to exit and re-enter D3PLOT, making the "write, test, modify, re-test" development cycle very quick and easy.



15.2.1. Compiling and Running a Script

Compiling and Running a script



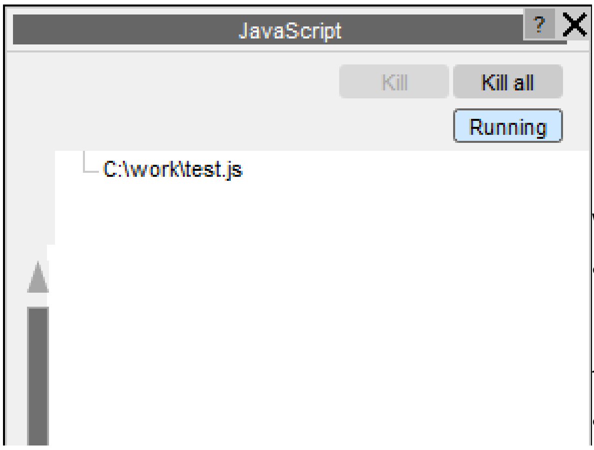
Run will both compile and run the script unless it contains syntax errors, in which case it stops with an error message when compilation fails.

Debug Starts the JavaScript debugger, JaDe to debug the script.

Check only compiles the script, reporting any errors found, and does not run it.

Encrypt A script can be encrypted so that the source code is hidden but the script can still be run (when compiling and running the script D3PLOT decrypts the file in memory). Once

	encrypted the source code cannot be retrieved by an ordinary user so make sure that you keep the original file somewhere safe. As a last resort contact OASYS Ltd who can decrypt the script if required. If a script is split up into separate files by Use the files are all combined together into the main file before encrypting.
Merge	If a script is split up into separate files by Use the files are all combined together into a single file. This may be useful if you want to give the script so someone else and you do not want to have to give lots of different files.
GUI Builder	Opens the GUI Builder to interactively build GUIs for your script.
Memory size	Is the threshold size of the JavaScript memory "arena" at which Garbage Collection will take place.
Running	Enables you to view (and kill) any scripts that are currently running



Knowing which scripts are running

When scripts are running in D3PLOT, they are shown in the Running submenu.

Use the **Running** button to toggle between the list of scripts that are currently running, and the tree of available scripts.

To kill/terminate a running script, highlight the script(s) in the tree and press **Kill**.

Alternatively, to kill all the scripts that are currently running use **Kill all**.

15.2.2. Dealing with Errors in Scripts

Dealing with Errors in Scripts

Script errors come in two forms:

`Syntax errors`	Are mistakes of JavaScript grammar or spelling, resulting in error messages during compilation. These are easy to detect and correct since the line number and offending syntax are both described by the compiler. The script needs to be edited to correct the problem and then recompiled. Sometimes several iterations of the compile/edit cycle are required to eliminate all errors from a script.
`Run-time errors`	Are errors of context or logic in scripts that are syntactically correct, and thus have compiled, but which fail at some stage when being run. A typical example of a run-time error is an attempt to divide a value by zero, yielding the illegal result infinity. More subtle errors involve passing an invalid value to a function, accessing an array subscript that is out of range, and so on. The JavaScript API Reference Manual has been written in such a way that it handles "harmless" run-time errors by issuing a warning and continuing execution, but that more serious errors which could result in the wrong answers being generated issue an error message and terminate.

Here is an example script which demonstrates both types of error. This script lists all the shell elements attached to the first node in the model, and calls to the JavaScript API are hyperlinked to their relevant function definitions.

Copy Code
JavaScript

```
if(i = GetElemsAtNode (1, SHELL))
{
Print ("Number of shell elements on node " + GetLabel (NODE, 1) +
" = " + i.nn + "\n");
```

```
for(j=0; j<i.nn; j++)
{
k = j + 1;
Print ("Shell #" + k + " = " + GetLabel (SHELL, i.list[j]) +
"\n");
}
}
else
{
Print ("No shells at node " + GetLabel (NODE, 1) + "\n");
}
```

This initial script is syntactically correct, and on an example model writes the following to the controlling terminal:

```
Number of shell elements on node 1 = 4
Shell #1 = 31318414
Shell #2 = 31318415
Shell #3 = 31319004
Shell #4 = 31319006
```

If a syntax error is deliberately introduced by omitting the second bracket at the end of line 1, leaving the "if" statement incomplete, ie:

Copy Code
JavaScript

```
if(i = GetElemsAtNode (1, SHELL)
```

Then this produces the compilation error:

```
Error when compiling J:\javascript\demo.js: at line 2:
SyntaxError: missing ) after condition
```

Which is computer-speak for "you left out the closing bracket on that '`if`' statement".

If a run-time error is deliberately introduced by omitting the second argument (**SHELL**) to GetElementsAtNode (), making the first line:

Copy Code
JavaScript

```
if(i = GetElemsAtNode (1))
```

Then this is not picked up during compilation because the syntax is correct, but shows up when the script is run with the message:

```
%%% ERROR %%%
Fewer than 2 arguments supplied to Javascript function
<get_elements_at_node>
```

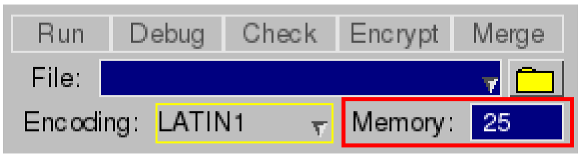
And the script run terminates prematurely.

Because this example script has only one call to GetElementsAtNode () it is easy to identify and correct the problem, but in more complex scripts with many such calls it may be necessary to insert diagnostic Print () statements in order to track down a particular error.

15.2.3. Setting the Garbage Collection Memory Size

Setting the Garbage Collection Memory Size

(This is an advanced topic, and you don't need to understand it.)



JavaScripts execute inside a memory "arena", allocated dynamically from the operating system, which grows in size as storage is requested within the script. This growth occurs due to requests for "new" variables within the script and also when API functions allocate and return values and objects, and it is limited only by what the operating system can deliver.

The nature of JavaScript means that objects frequently become redundant, and it is wasteful not to reuse the storage that they occupy, therefore there is a "Garbage Collection" process running behind the scenes which periodically checks storage and releases that which is no longer needed. This process is automatic and hidden from the user, it just "happens".

However Garbage Collection is quite a CPU-hungry process, so it is only carried out periodically when a certain threshold is reached. This can sometimes be observed during script execution as a periodic "pause for thought", and if you are monitoring memory usage with a system tool you may see it drop during these pauses.

Clearly this threshold value must be large enough not to trigger excessively frequent (and costly) garbage collections, while at the same time not being so large that scripts build up large amounts of excess memory to the detriment of the rest of the programme.

The Memory size value in the JavaScript panel is the amount of memory allocated for garbage collection. Every time a new object, array, string or double precision number is used a garbage collection 'thing' is also allocated. The Memory size is the total memory for these 'garbage collection things', **NOT** the total memory for the script. The total memory for the script could be significantly higher than this value. e.g the memory required for a Model object could be several kbytes but the memory for the 'garbage collection thing' for the Model object will something like 10 bytes for a 64bit operating system.

When the memory used for garbage collection 'things' reaches a significant proportion of Memory Size (normally about 2/3) then garbage collection will take place to try to reclaim memory. If no memory can be reclaimed and the total memory used for garbage collection reaches Memory size then the script will terminate with an error.

If your script has to retain a large number of objects, arrays, strings etc in memory then you may have to increase the value for Memory size . This can also be done using the `d3plot*javascript_memory_size` preference or adding a special memory comment at the top of the script.

To recap:

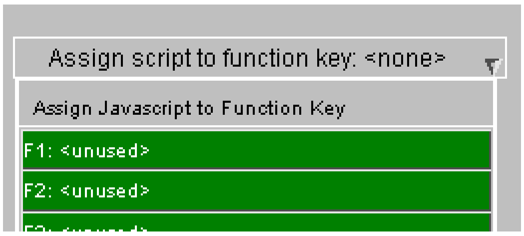
- This threshold does ***not*** limit the memory the script can use, that is limited only by the operating system.
- It sets the memory for Garbage Collection 'objects'.
- Scripts which allocate a lot of memory, and which exhibit frequent pauses, ***may*** run faster with a larger value.
- ... and finally:

  If you don't understand this topic don't worry. Most scripts will run quite happily with the default value, and you can ignore this setting unless they appear to be struggling, in which case try raising it. (As good an approach as any is to keep on doubling this value until the script works, but don't use very large sizes unnecessarily.)

15.2.4. Assigning JavaScripts to Function Keys

Assigning JavaScripts to Function Keys

If a script is to be run repeatedly it can be convenient to set up a short-cut to it by assigning it to a function key, giving a "one click" method of running it.



Function keys can also be used to run D3PLOT command (.tcf) files, see Utilities, Function Keys , and key assignment may mix the two types at will. JavaScripts so assigned should use the extension ".js" since this is how the two file types are distinguished from one another when the function key is used: files with any extension which is ***not*** ".js" are assumed to be command files.

Assignment of JavaScripts to function keys can also be saved in the oa_pref file in exactly the same way as command files see Utilities, Function Keys for details of how this is done.

15.2.5. Maintaining a Library of JavaScripts

Maintaining a library of JavaScripts

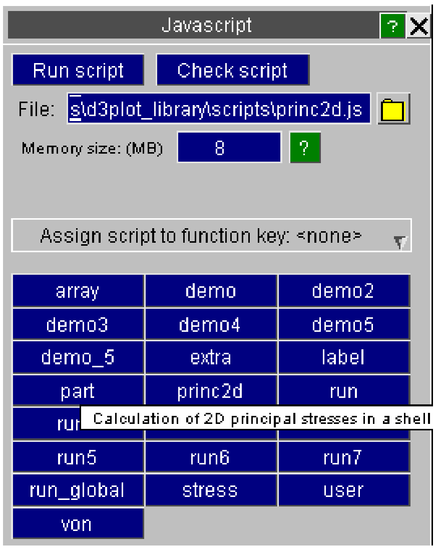
It is also convenient to have a library of scripts in a defined location.

By default D3PLOT looks in **`$OASYS/d3plot_library/scripts`** , but you can define a different directory by setting the preference:

`d3plot*script_directory:` *`some_different_directory_name`*

In the your oa_pref file.

All scripts found in the relevant directory will be listed in the JavaScript panel, as shown in this example.



Using the "description:" comment at the top of a script to identify its purpose

To help to identify scripts special comments are searched for in the top 10 lines of each script, and if **`description:`** is found, for example the comment line:

```
// description: Some description of the script's purpose
```

Then the description line is shown as hover text when the mouse is placed over that filename. In the example above the " **`princ2d`** " script has the line

```
// description: Calculation of 2D principal stresses in a shell
```

Using the "name:" comment at the top of a script to change its name

Normally the name shown for a script will be its filename, stripped of any leading pathname and trailing ".js" extension.

However if the string **`name:`** is found in the first ten lines of the script, then the following name will be used instead. For example the line:

```
// name: temporar y
```

Will result in the script appearing with the name " **`temporary`** " in tha JavaScript panel. This does not affect the actual name of the script, only the name on its library button.

Using the "memory:" comment at the top of a script to change the initial garbage collection memory

Sometimes the memory required for garbage collection needs to be changed.

If the string memory: is found in the first twenty lines of the script, then the size given will be used for the memory (unless the size in the memory textbox is larger than this value). For example the line:

```
// memory: 50
```

Will result in the script using 50Mb for garbage collection memory.

Using the "encoding:" comment at the top of a script to change the encoding

By default the encoding used for scripts is LATIN1

If the string **`encoding:`** is found in a comment on the first twenty lines of the script, then the encoding will automatically be used for the script. The allowed values are **`UTF8`** or **`UTF-8`** for UTF- 8 encoding and **`ShiftJIS`** , **`Shift-JIS`** or **`sjis`** for Shift-JIS encoding.

For example the line:

```
// encoding: UTF8
```

Will result in the UTF-8 encoding being used for the script.

Using the "module:" comment at the top of a script for ES6 modules

D3PLOT has to compile scripts that use ES6 modules differently to 'normal' scripts. If a script has the extension `.mjs` then D3PLOT will automatically compile the script to use ES6 modules. Alternatively, if the file has a different extension, the `module` comment can be used to tell D3PLOT that this file needs to be compiled to support ES6 modules.

If the string **`module: TRUE`** is found in a comment on the first twenty lines of the script, then the script will be compiled with ES6 module support.

For example the line:

```
// module: TRUE
```

Will result in the script being compiled with ES6 module support.

15.2.6. Running a JavaScript in "Batch" Mode

Running a JavaScript in "batch" mode

All the above assumes that JavaScripts will be run interactively from the user interface, however it is also possible to run a script in "batch" mode using the command line interface. The relevant command-line commands are:

/JAVASCRIPT -	**+-**	**COMPILE**	Compiles and checks the script, but does not run it.
	+-	**EXECUTE**	(Re)compiles and runs the script
	+-	**MEMORY <nnn>**	Resets the Garbage Collection threshold to <nnn> MBytes

To run a JavaScript from batch these commands need to be placed in a command file and run using the command line " **-cf=** *command filename* " option. For example the command file might be:

```
... some other commands
/JAVA EXEC my_script.js
...some further commands
```

And the command line required to run D3PLOT might be something like:

```
$OASYS/d3plot93.exe -d=default -cf= command_file -exit analysis_name
```

Obviously multiple script invocations may be placed in a command file. For more information see:

Command and Session files	Describes command files, and explains how to create and use them
Valid D3PLOT command line arguments	Describes the various command line arguments, and how to use them

15.2.7. ECMAScript 6 Modules

ECMAScript 6 modules

D3PLOT 21.1 supports ES6 modules. For more information on ES6 modules, please to https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules.

Prior to support for ES6 modules, the only way to split up a script in D3PLOT was to use the non-standard Use() functionality in the Oasys Ltd software. ES6 modules now give JavaScript built-in support for modular programming using the `import` and `export` keywords. D3PLOT supports both static and dynamic imports for modules and this section gives a brief guide to how D3PLOT locates modules.

To be able to support ES6 modules, D3PLOT has to compile the script in a different way to a 'normal' script that does not use modules. So that D3PLOT can tell how to compile the script we use a different extension `.mjs` for scripts that use modules. This follows the convention used by V8 and Node.js. Alternatively, if you prefer, you can put a special `module` comment at the top of the script and continue to use the extension `.js` (or whatever other extension you prefer).

When importing modules using import then if the module filename is an absolute filename D3PLOT can locate the file directly. However if a relative filename is used D3PLOT will search for the file in the following order.

- Relative to the directory that the main script is in
- Relative to any parent module directory
- Relative to the script directory specified in the OA_ADMIN directory. By default this will be $OA_ADMIN/d3plot_library/scripts but this can be changed with the script_directory preference in the OA_ADMIN oa_pref file
- Relative to the script directory specified in the OA_INSTALL directory. By default this will be $OA_INSTALL/d3plot_library/scripts but this can be changed with the script_directory preference in the OA_INSTALL oa_pref file
- Relative to the script directory specified in the HOME directory. By default this will be $HOME/d3plot_library/scripts but this can be changed with the script_directory preference in the HOME oa_pref file
- Relative to the current directory

- Relative to any script_directory preference specified in a preference file given by a `-pref=xxxx` command line option.

Note that the non-standard Use() functionality and ES6 modules cannot both be used when compiling a script. You must use one or the other. Where possible you should now use ES6 modules in preference as they are now part of core JavaScript.

Individual module files can be encrypted if required so if you want to protect only some parts of your code/process and leave the rest of it open/visible this can easily be done.

One difference between using the non-standard Use() method and ES6 modules is that with the Use() method D3PLOT could merge all of the individual files back into a single file using the **Merge** command which could then be encrypted if required to only have to give out a single file instead of a 'package'. For ES6 modules an external tool such as rollup.js or Webpack is required to merge the files. Once combined to a single file, D3PLOT can encrypt it.

15.2.8. Scripts using GUIs

Scripts using GUIs

Scripts that create a graphical user interface (GUI) have to keep running so that the windows for the user interface remain visible. The way that this is done has changed in version 21.

All programs that have a graphical user interface (GUI) use an "event loop" to process any mouse/keyboard events. D3PLOT has a main "event loop" to process all of the program's events.
In version 20 and before, if a script created and showed a window, D3PLOT would start a new "event loop" to manage and process that JavaScript window.
The script would not return from the window Show() call until the window was hidden/closed.

i.e. showing the window would "block" execution of the script until the window was closed.

When the window is closed, the script continues.
When execution reaches the end of the script, the script is terminated

For example, in version 20 and earlier, in the following script, "Hello, world!" will not be printed until the window is closed because the call to `w.Show()` will not return until the window is closed.

```
// Create a window with a widget
var w = new Window("Test", 0.5, 0.6, 0.5, 0.6);
var l = new Widget(w, Widget.LABEL, 0, 50, 0, 6, "Press X to close
the window");

// Show the window and start event loop
w.Show();

// Print message
Message("Hello, world!")
```

When the window is closed the message is printed and the script will then terminate as execution has reached the end of the script.

In version 21 the behaviour has changed. If a script creates and shows a window, D3PLOT will ***not*** start a new "event loop" to manage and process that JavaScript window.
The window will now be processed from the main event loop in D3PLOT.

The script now returns from the window Show() as soon as the window is shown, and execution of the script continues.

i.e. showing the window no longer "blocks" execution of the script until the window is closed.

When execution reaches the end of the script, the script is ***not*** terminated.

The script continues running "in the background" as the script has shown a window.

For example, in version 21, in the same script "Hello, world!" will be printed immediately after the window is shown, because the call to `w.Show()` returns after the window is shown.

When the message is printed, the script will ***not*** terminate when execution reaches the end of the script. The script will continue to run "in the background". You can use the Running button in the script menu to see which scripts are running at any time.

If a script that shows windows continues running "in the background", and does not terminate when execution reaches the end of the script, how/when does the script terminate?

A script that uses windows ***must*** now call **`Exit`** to terminate the script

```
// Create a window with a widget
var w = new Window("Test", 0.5, 0.6, 0.5, 0.6);
var l = new Widget(w, Widget.LABEL, 0, 50, 0, 6, "Press X to close
the window");

// Exit when window closed
w.onClose = Exit;

// Show the window
w.Show();

// Print message
Message("Hello, world!");
```

If the script does not call Exit then the script will continue to run in the background, even if no windows are being shown. If necessary the Running menu can be used to terminate the script.

This change was primarily required for running multiple scripts in PRIMER, but the behaviour in D3PLOT is the same. See Why the change has been made in version 21 in the PRIMER manual for details.

For simple scripts, the only change that should be required is to make sure that the script exits when the main window is closed by calling `Exit()`. The Window onClose event can be used to do this.

If you have specifically relied on script execution "blocking" when the Window `Show` method is used, then your script may need to be slightly modified. Please contact our support team if you have any questions or need assistance.

15.3. The D3PLOT JavaScript API

The D3PLOT JavaScript API

The API is documented in the JavaScript API Reference Manual

The global class section lists all the functions available in the global scope. To make them easier to locate they are also presented in different categories in the tree on the left hand side.

15.4. Examples

Examples

By far the easiest way to learn JavaScript is by example and, more specifically by modified existing scripts to do what you want.

The software comes supplied with examples in the **`$OASYS/`** *`programme_`* **`library/examples`** directory (for D3PLOT **`$OASYS/d3plot_library/examples`**) and you are free to use and modify these files for your own purposes.

There are also some simple documented examples in the JavaScript API reference manual.

16. More About Data and Data Components

16.1. Introduction to this Section on Data and Data Components

Introduction to this section on data and data components.

It is important that you understand what data LS-DYNA writes, and how D3PLOT processes it for presentation. Therefore this section is organised to provide the following information:

Format and contents of the LS-DYNA databases.

In Section 13.1 the format and in Section 13.2 the contents of each of the LS-DYNA databases processed by D3PLOT are described in more detail. The various output switches available in LS-DYNA to control their contents are also described.

Global and summary data descriptions.

D3PLOT makes a distinction between global and summary data components, (those for the whole model, materials and contact surfaces), and components for individual element types. Sections 13.3 to 13.5 describe the "summary" data components available for the whole model, materials and contact surfaces respectively.

Data components for nodes and element types that write results.

Not all entity types write results to the files processed by D3PLOT. Only nodes, solids, shells, thick shells, beams and contact surface segments have data components available for them. These are described in Sections 13.6 to 13.11 respectively.

Data components for entities that do not write results.

Those entities which do not write results to databases processed by D3PLOT are lumped-masses, springs, seat-belt types, joints and stonewalls. Nevertheless geometric and other data components are available for these and are described in Section 13.12.

Theory behind data manipulations.

The theory and formulae behind the computation of derived data components is given in Theory and Formulae . For example von Mises stress, principal stresses, etc.

16.2. Format of the LS-DYNA Databases Processed by D3PLOT

Format of the LS-DYNA databases processed by D3PLOT

D3PLOT processes three of the binary database file types from LS-DYNA:

Complete state
Dynamic relaxation
Eigenvalue (ex Nike)
	`.ptf`
	`.rlf`
	----- 	
		Contains basic geometry and topology of, and transient results for, nodes, solids, shells, beams and thick shells. Eigenvalue files contain modeshape (not transient) results.

Contact force	`.ctf`	Contains geometry and topology of, and transient results for, contact surface segments.
Extra time history	`.xtf`	Contains geometry and topology of lumped-masses, springs, seat-belt types, stonewalls and joints. (But the results for these are not processed by D3PLOT.) Note that the .xtf file is not supported by MPP dyna, and that it is increasingly likely to be supplanted by the "binout" (or LSDA) database file. From V9.0 onwards the `.ztf` file contains all the plottable data previously extracted from the .xtf file, so there is no loss of functionality in D3PLOT.
Pseudo time history	`.ztf`	Contains extra information culled from the input deck: nodal contacts, restraints and constrained, part and contact names. From V9.0 onwards also contains all static data previously saved in the `.xtf` file, making plotting of these extra items possible if an .xtf file is not present. The `.ztf` file is generated by PRIMER directly from the input deck, usually by running a batch translation phase immediately after the ls-dyna analysis.

16.2.1. The "Familied" Nature of Database Files

The familied nature of database files.

All of the database files above use a family structure. There is always a "root" member, which may have between 1 and 999 "children". This is done to keep file sizes down:

there are many advantages to having a few moderately big files instead of one huge one.

The "root" members:	Have the names `<job>.ptf` , `<job>.ctf` , etc.
The "child" members:	Have names `<job>.ptf01` to `<job>.ptf99` ; then `<job>.ptf100` to `<job>.ptf999`

The maximum size of a family member is set when LS-DYNA is run, and a new "child" member is opened if writing the current block of information would cause the current member to spill over this limit.

By default 7Mbytes is used (in single precision), corresponding to 1835008 words of data, but it is possible to change this when the LS-DYNA job is submitted via the X= parameter on the DYNA submission line, or via the Shell. However very large analyses often use a larger family size in order to stop results states spilling over into multiple family members.

By default D3PLOT determines the file family size automatically (by taking the greater size of the first two members of a family, and rounding up to the nearest Mbyte). You can over-ride this by setting an explicit size (see The FILE > popup menu options) but this should rarely, if ever, be necessary.

Note: Previous versions of D3PLOT used signed 32 bit integers to represent disk addresses, which limited total database size to 2^{31} words, or 2GWords.

D3PLOT 8.0 onwards uses 64 bit integers to represent disk addresses, even on 32 bit machines, which means that it can access disk addresses up to 2^{63} words, or 9e18 words.

LS-DYNA from approximately 2002 (~LS960) onwards no longer adheres to the maximum database size logic for .ptf files for larger models. If the state is too big to fit into a single family member it is allowed to increase in size in order to contain the state. D3PLOT handles this automatically.

16.2.2. Setting the Family Member Size of Database Files

Setting the family member size of database files

The following strategy is adopted when the software is supplied to you by Oasys Ltd:

- The whole software suite defaults to 7MByte family members if no external environment variables are set.

- The environment variable **`FAM_SIZE`** may be set to a family size in MBytes. This should be an integer between 1 and 100, and it is the preferred method of changing this value since it will be picked up by all the software in the suite.
- In the Shell you can set a different size when submitting a LS-DYNA job with the **Binary file size** slider in the **ADDITIONAL FILES** panel.
- In D3PLOT the value defaults to **`FAM_SIZE`** if this has been set, otherwise it is computed automatically. You can also modify the value both at file input and during a session using the commands described in Open a Single Model and The Animation options popup menu .

Please contact Oasys Ltd if you need more advice on changing this value.

16.2.3. Handling Missing Family Members

Handling missing family members

It is not possible to use any of the database files above if their "root" family member is missing. This contains control, geometry and topology data that is required if the rest of the file family is to be read. However it is possible to process families in which some children have been deleted.

The data written to a file family is: <Control and topology> <state #1> <state #2> ...

and certain rules are used to make selective removal of child family members easier.

(1) If adding a <state> would overflow the maximum size permitted in the current member, it is closed and a new one is opened.

(2) A new family member is **always** started following the writing of a restart dump file.

(3) If a <state> is too big to fit into a single member then the maximum size rule is still obeyed: the first member is filled to capacity, then the next one is opened to take the remainder of the data. Thus <states> will be written in pairs (or, if large enough, triplets, quadruplets and so on) of family members.

This is best explained by example. Two are given here: one for a moderate size analysis, and one for a huge one. In both cases a maximum family size of 7MB is assumed.

Example 1 : Basic control and topology requires 2MB, each state uses 3MB.

Root member		Child #3	Restart dump => new member
Control + Geometry	2MB	<State #5>	3MB
<State #1>	3MB	<State #6>	3MB

Child #1		Child #4	
<State #2>	3MB	<State #7>	3MB (Last state in file)
<State #3>	3MB		
Child #2			
<State #4>	3MB		

Example 2 : Basic control and topology require 8MB, each state 12MB.

Root member		Child #2		Child #4	
Control + Geometry	7MB	<State #1: part 1>	7MB	<State #2: part 1>	7MB
(part 1)					
		Child #3		Child #5	
Child #1		<State #1: part 2>	5MB	<State #2: part 2>	5MB
Control + Geometry	1MB			And so on in pairs	
(part 2)					

Note that the basic geometry spills into the first child, and that subsequent <states> always come in pairs of files. In this example child family members containing states could be removed, but only in matched pairs.

Hint:

Use the UNIX command `ls -lt` to look at your files. This will give a "long" listing showing file size, and also sort them into chronological order of creation.

Then look for matched pairs of files that will, in this example, have 7MB and 5MB sizes with the smaller file being marginally more recent.

D3PLOT will skip gaps in a file family sequence if the `FILE_SKIP` environment variable or the D3PLOT preference file_skip is set. This is an integer that defines how many missing files will be skipped before the search is abandoned. The default value set for the preference and environment variable in the Shell is 50, but values much larger than this (up to 999) could be used. Larger values will increase the time delay when **SCAN** ning files as children are searched for on disk. See Open a Single Model and The FILE > popup menu options for ways to alter this value at run time.

16.2.4. Disk Format of Binary Database Files

Disk format of binary database files.

All the files above are "random access binary files". This means that they are stored in the binary format of the machine, and cannot be read or edited with tools such as a text editor.

There are several machine formats available, the most common one being:

IEEE	(Stands for the Institute of Electrical and Electronic Engineers). This is the most common, and a de-facto standard on Unix workstations.
CRAY	Used on Cray XMP, YMP, C90, J90, etc.
Convex	This is now no longer current, but older C1xx, C2xx and C3xx machines used this format.

There are two word-lengths in common use:

32 bit	Single precision, using 4 bytes. This gives 7 or 8 decimal figures of precision, and a decimal exponent in the range +/-38.
64 bit	Double precision, using 8 bytes. This gives 14 or 15 decimal figures of precision, and a decimal exponent of +/-308 in IEEE format, or +/-2465 in Cray format.

There are two possible way to arrange the bytes in the words:

Big endian (or **M** ost **S** ignificant **B** yte : **MSB**)	Byte order [4] [3] [2] [1] (single precision example)
Little endian (or **L** east **S** ignificant **B** yte : **LSB**)	Byte order [1] [2] [3] [4] (single precision example)

LS-DYNA is normally supplied such that it writes 32 bit IEEE (Big endian) files, regardless of the machine architecture and precision it is running on. This is achieved by converting results if necessary in the output routines.

D3PLOT is capable of reading any of the formats above. It detects the format by scanning the contents and converts it automatically on input to the native format of the machine on which it is running. Therefore you will sometimes see a message like:

```
[This machine is Cray 64 bit Normal endian ]

[File format is IEEE 32 bit Normal endian ]
```

when it reads files. This is just a notification message and you need not take any further action: in the example above they will be converted automatically to 64 bit Cray format.

Does using 32 bit format on a 64 bit machine reduce the precision of my answers?

It does not affect the precision of the ***calculation*** at all: this is always performed at the full precision of the machine, and dump files also use the full precision (so restarts do not compromise accuracy). It is ***only*** the databases written for post-processing that are truncated to 32 bits.

So yes, it does affect the precision in post-processing: you will be forsaking 14 or 15 significant figures for 7 or 8. But consider for a moment what this means: a typical car model might be 5 metres long, so single precision output will resolve displacements to somewhere between 0.05 and 0.5 microns. And anyway, is your calculation really accurate to 0.00001%?

It is hard to envisage the situation where this matters and, in the opinion of Oasys Ltd, the benefits of halving disk space usage and having files which may be post-processed without conversion on workstations far outweigh any disadvantages. However Oasys Ltd will supply the code with 64 bit output on request.

Are there any other limitations in using 32 bit file formats?

Only one: if you are post-processing on a 64 bit machine you will have problems if your 32 bit file contains integers (eg node, element or material numbers) outside the range $+/-2^{24}$ (16,777,216). This is because the bit patterns of integer and floating point numbers become hard to tell apart at this point, and this distinction is required for automatic conversion.

So, if you run on (say) a Cray ***and*** post-process on that machine then try to keep your node, element, material and other labels below this limit. If you take your files to a 32 bit machine for post-processing the problem will not arise and you can use the full valid integer range.

Alternatively, please request a version that writes 64 bit output from Oasys Ltd.

Why does D3PLOT sometimes mix up IEEE and Cray 64 bit formats in older files?

Prior to LS-DYNA 940 it was not possible to tell if a 64 bit file came from a Cray or an IEEE machine. (Subsequent database files contain values which make distinction possible.)

Therefore if the file type is indeterminate D3PLOT defaults to one or the other format, which may be wrong. You can control the default 32 and 64 bit file types using the following environment variables:

Variable name	Word size	Possible values	Default in V8.0
FILE_TYPE_32	32 bit	IEEE, CONVEX	IEEE
FILE_TYPE_64	64 bit	CRAY, IEEE, CONVEX	IEEE (was CRAY in V7.x)

e.g. **setenv FILE_TYPE_64 CRAY** Would be required to read an older 64 bit Cray file in V8.0

16.3. Contents of the LS-DYNA Database Files Processed by D3PLOT

Contents of the LS-DYNA database files processed by D3PLOT

(See Key to finding information in LS-DYNA database files for a summary table of all contents in all files.)

16.3.1. The "Complete State" (.ptf) File (Also .rlf and d3eigv Files)

The "complete state" (.ptf) file. (Also .rlf and d3eigv files.)

Defining output parameters in LS-DYNA

The following LS-DYNA control cards control output of this file:

*DATABASE_BINARY_D3PLOT	Is mandatory. Controls output frequency.
*DATABASE_EXTENT_BINARY	Is optional. This card allows switching of certain parts of the file's contents on/off. These are discussed in Controllable contents of the complete state file below.

The generic contents of the complete state file

Control information:

Number of nodes

Number of each type of element

Number of materials

Amount of data written for each element type

Basic geometry & topology:

Undeformed nodal coordinates

Element topology

Arbitrary numbering tables

Complete state: (Repeated for all times dumped)

Time

Global data: energies, velocities, masses, stonewall forces

Nodal coordinates

Temperatures at nodes

Nodal velocities

Nodal accelerations

Solid element stresses and plastic strain

Thick shell stresses and plastic strains

Beam forces

Thin shell stresses, strains & related data

Deleted element tables

Controllable contents of the complete state file.

The options on the **`*DATABASE_EXTENT_BINARY`** card in the LS-DYNA input deck allow you to control the following contents of the complete state file.

`NEIPH` "Extra" data components for solid elements. Default **None** .

Some material models generate more information for solids than can be written in the standard formats. For these "extra" variables may be written. They are <neiph> scalar values that follow the normal data. They:

- Are written for every solid element, regardless of material model.
- Have no explicitly defined component names associated with them: this will depend on the material model.

NEIPS "Extra" data components for shells and thick shells. Default **None** .

As with solids some material models write extra information for shells. They are **`<neips>`** scalar values that:

- Are written at every "surface" output for every shell, regardless of material model used.
- Have no explicitly defined components associated with them: this will depend on the material model.

MAXINT Number of "surfaces" written for shell and thick shell elements. Default **`3`** .

The default value of 3 writes data at neutral axis, innermost and outermost integration points for shells and thick shells. Values other than 3 write results for the first **`<maxint>`** integration points: ***read Section 13.8.2.2 before using them.***

STRFLG Write directional strain tensors for solids, shells & thick shells. Default **Off** .

By default no strain tensors are written for any elements, (although effective plastic strain is). Turning this flag on causes the strain tensors for solids, shells and thick shells to be written. Note that:

- A single tensor at the element centre is written for solids.
- Tensors at innermost and outermost integration points ***only*** are written for shells and thick shells, regardless of the **`<maxint>`** value.

The following flags can be used to reduce database size by controlling the output for shells and thick shells. By default they are all **On** .

NOTE: None of the following four flags influences the output for solid elements.

SIGFLG Controls the output of the stress tensors for shells and thick shells.

The symmetric stress tensor (6 values) is written at **`<maxint>`** "surfaces" for every shell and thick shell, so turning this off usually saves 18 values per shell. You will, of course, then not be able to post-process any stresses.

EPSFLG Controls the output of effective plastic strain for shells and thick shells.

The effective plastic strain (1 value) is written at <maxint> surfaces for every shell and thick shell. Turning this off will usually save 3 values per shell.

RLTFLG Controls the output of force and moment resultants for shells.

Force and moment resultants (8 values) are written for every shell (but not for thick shells). Turning this off will save 8 values per shell.

ENGFLG Controls the output of thickness and strain energy density for shells.

Thickness, strain energy density and two other (unused) values are written for shells (but not for thick shells). Turning this off will save 4 values per shell.

CMPFLG Composite material stress output in local axes. Default **Off** .

By default all stress tensors are written in the global coordinate system. Turning this flag on causes those from composite materials to be written in the material local axis system(s).

WARNING: ***There is no way for D3PLOT to tell from the database that these stress tensors are in the local system. It assumes that ALL STRESS TENSORS ARE IN THE GLOBAL SYSTEM.***

If you use this facility IT IS YOUR RESPONSIBILITY to interpret your results correctly: they will be reported in D3PLOT as global stresses.

BEAMIP Number of "extra" data values written for beam elements. Default **None** .

All beam elements write 6 basic forces and moments, but certain element and material formulations can generate extra data:

Resultant formulation (Belytschko-Schwer) beams can generate a further 15 "plastic" values, depending upon the material model used.

Integrated (Hughes-Liu) beams can generate 5 "extra" stress and strain values at each integration point.

There is no way of telling from the database whether "extra" beam results contain "Integrated" stress/strain data, or "Resultant" plastic data.

LS-970 and earlier: inconsistent beam sign conventions.

There is an inconsistency of sign convention in beam output from versions of LS-DYNA prior to release 971.

• "Resultant" (typically Belytschko-Schwer) elements use one sign convention
• "Integrated" (typically Hughes-Liu) elements use the opposite sign convention for 4 of the 6 output components.

The following table shows the status quo ***up to and including LS-DYNA release 970*** :

Component	Matching?
Fx	Same
Fy	Opposite
Fz	Opposite
Mxx	Opposite
Myy	Opposite
Mzz	Same

Which is right?

Sadly there is no "right" for beam output, as different users have different conventions. The confusion arises because of the different ways in which the beam types work: integrated beams have integration points at their centre, whereas resultant beams have (potential) hinges at their ends. The former reports force in the beam, and the latter reactions at the supports.

This has serious implications when plotting beam data and when extracting cut section forces and moments through beam structures (see FORCES Computing Forces and Moments on the Cutting Plane)

LS-971 onwards: beam sign conventions corrected.

At some stage during the development of LS-DYNA 971this bug was corrected, and the output from resultant and integrated beams now match. The sign convention that has been adopted is the "integrated" one.

D3PLOT handling of Beam sign convention problems.

Unfortunately it is not possible to determine the analysis code version accurately from a database file (at the time of writing, September 2008, output databases from LS-DYNA 971 still report their version number as 970). Therefore D3PLOT adopts the following approach:

- Contour plots of beam data are always shown "as is". This tends not to matter since it is usually easy enough to see what plots mean.

- However when computing cut-section forces through beams the sign convention is vital, therefore D3PLOT will prompt you for the sign convention to be used.

Update : From approximately 2009 onwards output from LS-DYNA 971 reports its version number correctly in output files, and D3PLOT is thus able to determine that the sign convention problem described above has been fixed.

IT IS YOUR RESPONSIBILITY to interpret your beam results correctly.

`DCOMP`	Data compression flag. Default **1.** The default value of "1" means that database compression is turned off and a full set of data values is written to the database for each rigid element in the model. If this option is set to "2" data values will not be output for rigid elements. This option can significantly reduce the size of the binary files written by LS-DYNA if the model contains a large proportion of rigid elements.

The following output options are currently not supported by D3PLOT, and should not be changed from their default (off) states:

`SHGE`	Output of shell hourglass energy.
`STSSZ`	Output of shell element time steps.

16.3.2. The Extra Time History (.xtf) File

`The Extra Time History (.XTF) file`

Defining output parameters in LS-DYNA

The following LS-DYNA control cards control output of this file:

`*DATABASE_BINARY_XTFILE`	Is mandatory. It controls output frequency and if you don't have this card you won't have an `.XTF` file.

The contents of this file are determined by LS-DYNA, and you have no control over them.

It is intended primarily for time-history processing, but D3PLOT uses it to obtain the geometry and topology of springs, seat-belt types, joints, stonewalls and lumped masses. D3PLOT ignores the state information in it (use T/HIS to process that).

`Generic contents of the .XTF file.`

Control data:

Title

Number of nodes

Number of elements

Amount of data written for elements

Initial data block:

Nodes for reactions

Spring topologies

Seat-belt etc topologies

Lumped mass data

Joint geometry

Stonewall geometry

Arbitrary numbering tables

Beam element time history blocks

Complete states (repeated n times)

(Not processed by D3PLOT)

16.3.3. The Contact Force (.ctf) File

16.3.3.1. Defining Output Parameters in LS-DYNA

Defining output parameters in LS-DYNA

The following LS-DYNA control cards control output of this file:

`*DATABASE_BINARY_INTFOR`	Allows you to set an output frequency for this file different to that used for the `.PTF` file, which is the default. (Note that you must request this file when submitting the job in order for it to be written.)

It is strongly recommended that you DO NOT use an output frequency for this file that is different to that used for the `.PTF` file. If you do D3PLOT can have problems trying to synchronise contact surface data with other results.

This file contains the topology and results for all segmented contact surfaces in the model. You have no control over its internal structure.

`Generic contents of the .CTF file`

Control information:

- Number of nodes
- Number of interface surface and segments

Geometry and topology:

- Undeformed nodal coordinates
- Interface segment topology

Arbitrary numbering tables

Complete state data: (repeated n times)

- Time
- Global data: Energies and velocities
- Current nodal coordinates
- Nodal velocities
- Interface segment stresses
- Nodal contact forces

16.3.4. Key to Finding Information in LS-DYNA Database Files

Key to finding information in LS-DYNA database files

	`.PTF file`	`.CTF file`	`.THF file`	`.XTF file`

Whole model data:

Global energies	KE, IE, TE	KE, IE, TE	KE, IE, TE	KE, IE, TE
Global velocities	VX, VY, VZ	VX, VY, VZ	VX, VY, VZ	VX, VY, VZ
External work				EW
Time-step			DT	

Data by material:

Material energies	KE, IE		KE, IE	
Material velocities	VX, VY, VZ		VX, VY, VZ	
Masses	Mass		Mass	

Nodal data:

Geometry	[X,Y,Z] coords	[X,Y,Z] coords	[X,Y,Z] coords	[X,Y,Z] coords
Displacements	[dX,dY,dZ]	[dX,dY,dZ]	[dX,dY,dZ]	[dX,dY,dZ]
Temperatures	[temps]		[temps]	
Velocities	[vX,vY,vZ]	[vX,vY,vZ]	[vX,vY,vZ]	
Accelerations	[aX,aY,aZ]		[aX,aY,aZ]	
Nodes for reactions				Geom + reacts

Solid & thick shell element data:

Topology	Connectivity & mat'l		Connectivity & mat'l	
Stresses &c	Stress tensor		Stress tensor	
	Plastic Strain		Plastic strain	
	Strain tensor		Strain tensor	
	Extra variables		Extra variables	

		(Els in TH blocks)	
Thin shell element data:			
Topology	Connectivity & mat'l	Connectivity & mat'l	
Stresses &c	Stress tensor	Stress tensor	
	Plastic Strain	Plastic strain	
	Strain tensor	Strain tensor	
	Extra variables	Extra variables	
	Force/moment resultants	Force/moment resultants	
	Thickness & strain energy density	Thickness & strain energy density	
		(Els in TH blocks)	
Beam element data:			
Topology	Connectivity & mat'l	Conn'ty & mat'l	Conn'ty & mat'l
Results	Forces & moments	Forces & moments	Extra data Mat'l 29
	Extra data Mat'l 29	(Els in TH blocks)	(Els in TH blocks)
Spring element data:			
Topology			Conn'ty & mat'l

Results				Force/elongation Moment/rotation
Seat-belt etc elem data:				
Topology				Conn'ty & mat'l
Results				Force/elongation /pullout etc
Lumped-mass element data:				
Topology				Node no & mass
Stonewall data:				
Geometry				All geom
Normal forces	Fwalls		Fwalls	Fwalls
Interface data:				
Geometry		All geom		
Summary forces				Fx,Fy,Fz summaries
Detailed results		Segment stresses All nodal forces		
Joint data:				
Topology				Nodes, type

		stiffness
Airbag (control volume) data:		
Results		Pressures etc

Notes on file types:

`.PTF` file :	"Complete state" file	is written at low frequency and has a complete description of the whole model.
`.CTF` file :	"Contact force" file	is written at the same frequency as the `.PTF` file, & is a complete description of the interface forces & stresses.
`.THF` file :	"Time history" file	is written at a high frequency and contains a (user-specified) subset of the model for detailed examination. It is used for XY graph type display (eg T/HIS).
`.XTF` file :	"Extra time-hist" file	is written at the same frequency as the `.THF` file, and contains extra information that cannot be accomodated in the `.THF` file structure. Used for XY plots, and also for providing the geometry for the graphics display of walls, joints, springs, etc
`.ZTF` file :	"Extra static data" file	is not generated by LS-DYNA, but rather by running PRIMER either in batch (immediately after the analysis) or manually. Extra "static" data are extracted from the keyword input deck and become available for post-processing.

16.4. Global (Whole Model) DataComponents

Global (whole model) data components

This section describes the data components available for the "whole" model. These can only be processed as numerical values in **WRITE** and **XY_DATA** .

Global data components are all extracted from the complete state (`.PTF`) file.

Raw components written by LS-DYNA are shown **THUS** , those calculated or derived by D3PLOT are shown oblique ***THUS*** .

Energies

KE_KINETIC_ENERGY	The total kinetic energy.
IE_INTERNAL_ENERGY	The total strain and other non-kinetic energy.
TE_TOTAL_ENERGY	The sum of the two terms above.

Velocities

VX_X_VELOCITY	Average X velocity of the whole model
VY_Y_VELOCITY	Average Y velocity of the whole model
VZ_Z_VELOCITY	Average Z velocity of the whole model
VR_VELOCITY_RESULTANT	Vector sum of above (computed by D3PLOT)

Mass values

MASS	The sum of all material masses.
Note:	This only contains mass from solid, shell, beam and thick shell materials. Lumped masses, stonewalls, and any other entity types which might contribute mass to the model are not included. Thus, for most models, this is an underestimate.

Momentum values (From mass * velocity)

MX_X_MOMENTUM	Mass * average X velocity
MY_Y_MOMENTUM	Mass * average Y velocity
MZ_Z_MOMENTUM	Mass * average Z velocity
MR_MOMENTUM_RESULTANT	Vector sum of the above

Note: The mass used here is the computed mass above, so the computed momenta may be an underestimate.

16.5. Part ("Material") Data Components

Part ("material") data components

This section describes the data components available for materials. These can be processed as numerical values in **WRITE** and **XY_DATA** , and from release 9.3 onwards part-based data may also be plotted as well. (In this case all elements of the part are "contoured" in the same colour representing their part value.)

Material data components are all extracted from the complete state (`.PTF`) file.

16.5.1. The "Material" Data Available Depends Upon the Files Present

The "material" data available depends upon the files present

Complete state files (.ptf)

These files are always present, and contain part data for:

1. All parts (materials) of solid, shell, thick shell, beam and SPH elements.
2. All nodal rigid bodies
3. All discrete and seatbelt elements.

Category (1) may be extracted and processed numerically (in **WRITE** and **XY_PLOT**), and visually (in 2D3D plotting) because the part numbers of these element classes are known.

Category (2), nodal rigid bodies, may be listed as a category under **WRITE, GLOBAL** ; however they cannot be plotting because their underlying nodes are not known.

Category (3), discrete and seatbelt element part data, cannot be processed from .ptf files alone because these files do not contain information about these elements.

ZTF files

If a .ztf file has been written from PRIMER then it becomes possible both to visualise discrete and seatbelt elements, and also to determine their part numbers. Therefore it is possible to associate part data with elements for these types, and process them in the same way as solids, shells, etc.

This also means that discrete and seatbelt elements can be selected and processed "by part" for operations such as blanking, picking, and so on.

XTF files

If an .xtf file is present, which will increasingly not be the case, as it is not supported by MPP versions of LS-DYNA, then discrete and seatbelt elements can be visualised; but the file does not contain the information required to associate these element types with their part ids as defined in the LS-DYNA keyword input deck, so their part-based data cannot be accessed.

D3PLOT automatically "culls" from part listings, menus, etc those parts which either do not contain any elements or for which the element type cannot be determined. The only time that they become visible is when a **WRITE** , **MATERIAL_SUMMARY** listing is produced.

16.5.2. The Data Components Available for Materials

The data components available for Materials

Raw components written by LS-DYNA are shown **THUS** , those calculated or derived by D3PLOT are shown oblique ***THUS*** .

(Note that these are the same component names as are used for global (whole model) results.)

Energies

KE_KINETIC_ENERGY	The material kinetic energy.
IE_INTERNAL_ENERGY	The material strain energy.
TE_TOTAL_ENERGY	The sum of the two terms above.

Velocities

VX_X_VELOCITY	Average X velocity of the material
VY_Y_VELOCITY	Average Y velocity of the material
VZ_Z_VELOCITY	Average Z velocity of the material
VR_VELOCITY_RESULTANT	Vector sum of above (computed by D3PLOT)

Mass values

MASS	The reported material mass.

Momentum values (From mass * velocity)

MX_X_MOMENTUM	Mass * average X velocity
MY_Y_MOMENTUM	Mass * average Y velocity
MZ_Z_MOMENTUM	Mass * average Z velocity
MR_MOMENTUM_RESULTANT	Vector sum of the above

16.6. Contact Surface Summary Components

Contact Surface summary components

This section describes the summary (whole surface) data components available for contact surfaces, as opposed to those available for contact segments (which are described in Contact segment results).

These cannot be plotted, they can only be processed as scalar values in **WRITE** and **XY_DATA** .

Contact surface summary data components are all extracted from the contact force (`.CTF`) file. They are all forces, computed in D3PLOT by summing the nodal forces on each surface.

16.6.1. The Data Components Available

The data components available

Forces

FX_CONTACT_X_FORCE	Total X force on the surface.
FY_CONTACT_Y_FORCE	Total Y force on the surface.
FZ_CONTACT_Z_FORCE	Total Z force on the surface.
FR_CONTACT_FORCE_RES	Vector sum of above.

16.6.2. Why Results from the .ctf File May Differ from Those in the .xtf File

Why results from the .CTF file may differ from those in the .XTF file

The `.CTF` file is an instantaneous "snapshot" of the status of all contact surface segments at the time-step when the dump took place.

The `.XTF` file output is generated within LS-DYNA by averaging contact forces on each surface over the time period since the previous time-history dump. Therefore the results are not instantaneous.

There are pros and cons to both approaches: the averaging used for `.XTF` output means that very fast spikes are not missed, but their magnitude is attenuated by the averaging process; whereas the instantanous data in the `.CTF` file reports the correct magnitude but, in doing so, may miss a spike altogether. The **.XTF** approach is better for time-history plotting, and the `.CTF` approach for graphical output.

16.6.3. When the A and B Side Forces Differ

When the A and B side forces differ

Where the forces on A and B surface sides are with 5% of one another the average value is reported. Where the difference is greater the side with the higher force magnitude is used.

Differences can occur - normally one side is zero, for example in the surface of single-surface contacts, or discrete nodes impacting a surface. (In the latter case the nodes are not on segments, so are not written to the `.CTF` file.)

16.7. Nodal Data Components

Nodal data components

LS-DYNA writes nodal results to the `.PTF` and `.CTF` files. All directional results are in the global cartesian system except accelerations at nodes classified as accelerometers.

D3PLOT is able to plot nodal data on 2D and 3D elements by averaging across their faces.

It will process scalar nodal data, element data averaged at nodes and other geometrical data in the **WRITE** and **XY_PLOT** menus.

16.7.1. The Nodal Data Components Available from the .ptf File

The nodal data components available from the .PTF file

Undeformed coordinates

BX_BASIC_X_COORD BY_BASIC_Y_COORD BZ_BASIC_Z_COORD

Current coordinates

CX_CURRENT_X_COORD **CY_CURRENT_Y_COORD** **CZ_CURRENT_Z_COORD**

Displacements (Derived from <current> - <undeformed>)

DX_X_DISPLACEMENT **DY_Y_DISPLACEMENT** **DZ_Z_DISPLACEMENT**

DR_DISP_RESULTANT

Velocities

VX_X_VELOCITY **VY_Y_VELOCITY** **VZ_Z_VELOCITY**

VR_VEL_RESULTANT

Accelerations

AX_X_ACCELERATION **AY_Y_ACCELERATION** **AZ_Z_ACCELERATION**

AR_ACCEL_RESULTANT

Relative Rotations (Derived from <current> and <undeformed>)

RDXR_X_ROTATION_REL **RDYR_Y_ROTATION_REL** **RDZR_Z_ROTATION_REL**

RDRR_R_ROTATION_REL

Relative rotation is calculated as the rotation of the position of a node, i.e. ignoring any nodal rotation, about the defined axis. The axis is taken from the reference node definition (see REFERENCE_NODES Calculating results with respect to one or three nodes).

If no reference node definition is switched on then the relative rotations will be calculated about the global axis directions through the origin.

If a single reference node is defined then the relative rotations will be calculated about the global axis directions through the reference node.

If three reference nodes are defined then the axis about which relative rotations are calculated depends on whether results are set to be in "Global" or "Local (N1..3)" on the Ref node panel. With "Global" set behaviour is the same as for a single reference node, i.e. relative rotations will be calculated about the global axis directions through N1. With "Local (N1..3)" set relative rotations will be calculated about the local axis directions through N1.

16.7.1.1. Temperatures

Temperatures

Note: Temperatures are only written in a thermal-only, combined structural and thermal, or structural analyses using thermal materials.

TEMPERATURE

Coordinates, velocities and accelerations are not written in a thermal-only analysis.

If "per surface" temperatures are present in the database, see below, then the simple **TEMPERATURE** data component reports the middle surface result.

Depending upon the value of THERM on the *DATABASE_EXTENT_BINARY card models may also contain:

TFX_TEMP_X_FLUX
TFY_TEMP_Y_FLUX
TFZ_TEMP_Z_FLUX

TFM_TEMP_FLUX_MAGNITUDE

TB_BOTTOM_TEMP
TM_MIDDLE_TEMP
TT_TOP_TEMP

These bottom, middle and top temperatures refer to the relevant surface of **SHELL** elements, but are written out at **NODES** . Nodes on solid elements have the same temperature value repeated at all three surfaces, the output for nodes on thick shell elements is not known at the time of writing.

Since these "per surface" temperatures are nodal data, and also since shell output may be present in the database for some number other than these 3 surfaces (see MAXINT on *DATABASE_EXTENT_BINARY) it would be misleading to use the normal "shell surface" selection method since that could imply that temperatures are available at all integration points. Therefore these are treated as unrelated nodal quantities, and in order to see temperatures at a given shell surface it is necessary to select the relevant component explicitly.

If flag DTDT on *DATABASE_EXTENT_BINARY is set then the following temperature component will be output

TR_TEMP_RATE The rate of change of temperature, dTemp/dTime.

Other nodal data

If MSSCL on *DATABASE_EXTENT_BINARY is set:

MASS_SCALING	Either incremental or %age change of nodal mass. (There is no way to tell which it is from the information in the database file, you will have to interpret it correctly.)

16.7.2. The Nodal Components Available from the .ctf File

The nodal components available from the .CTF file

Contact Forces

XG_GLOBAL_X_FORCE

FM_FORCE_MAGNITUDE

YG_GLOBAL_Y_FORCE ZG_GLOBAL_Z_FORCE

Local contact forces use the average local axis system of the parent segments meeting at the node. This is only an approximation.

XL_LOCAL_X_FORCE ***YL_LOCAL_Y_FORCE*** ***ZL_LOCAL_Z_FORCE***

Note: Contact force results are only available if a `.CTF` file is available, and forces are only available at nodes which form part of the topology of contact segments. For example forces at the "discrete nodes" which impact a surface, or nodes "spot-welded" to a surface, are not available.

16.7.3. Element Data Components Averaged at Nodes by D3PLOT

Element data components averaged at nodes by D3PLOT

Where nodes are attached to underlying elements for which averaging at nodes is meaningful, (solids, shells, thick shells and contact segments), D3PLOT will report averaged values for scalar output.

The averaging logic is the same as that required to produce contour plots with one important exception: contouring a facet provides a "parent" element for the node, so that discontinuities between elements meeting at the node can be resolved by reference to the parent. However averaged data at nodes for scalar output cannot have such a "parent", and some logical errors can occur: for example averaging data where a node is common to elements of two different types.

Averaged STRESS data available at nodes

(Solids, shells, thick shells)

The basic tensor components:

$$\begin{bmatrix}\sigma_{xx} & & \\ \tau_{yx} & \sigma_{yy} & \\ \tau_{zx} & \tau_{zy} & \sigma_{zz}\end{bmatrix}$$

X_DIRECT_STRESS	**XY_SHEAR_STRESS**
Y_DIRECT_STRESS	**YZ_SHEAR_STRESS**
Z_DIRECT_STRESS	**ZX_SHEAR_STRESS**

And the stresses derived from these by D3PLOT (Manipulations of stress tensor components):

MAX_PRINC_STRESS	***MAX_DEV_PRINC_STRESS***	***VON_MISES_STRESS***	***PRESSURE***
MID_PRINC_STRESS	***MID_DEV_PRINC_STRESS***	**SIGNED_VON_MISES_STRESS**	***TRIAXIALITY***
MIN_PRINC_STRESS	***MIN_DEV_PRINC_STRESS***	***MAX_SHEAR_STRESS***	***LODE_ANGLE***
YIELD_UTILISATION_FACTOR	***YIELD_UTILISATION_PERCENTAGE***	***LODE_PARAMETER***	***LODE_PARAMETER_ALT***

Averaged STRAIN data available at nodes

(Solids, shells, thick shells)

The effective plastic strain: (see A further explanation of strain components for more information about strains)

STRAIN

$$[\epsilon_p]$$

The basic strain tensor:

SX_DIRECT_STRAIN	**SXY_SHEAR_STRAIN**

SY_DIRECT_STRAIN **SYZ_SHEAR_STRAIN**

SZ_DIRECT_STRAIN **ZX_SHEAR_STRAIN**

$$\begin{bmatrix} \epsilon_{xx} & & \\ \epsilon_{yx} & \epsilon_{yy} & \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix}$$

And the strains derived from these by D3PLOT (Manipulations of Strain Tensor Components):

SMAX_PRINC_STRAIN ***SVON_MISES_STRAIN*** ***SAV_AVERAGE_STRAIN***

SMID_PRINC_STRAIN ***SMAX_SHEAR_STRAIN***

SMIN_PRINC_STRAIN ***PEMAG_PLAST_STRN_MAG***

The 1st 99 extra data components (if present)

SO1_SOLID_EXTRA_1 to **SO9_SOLID_EXTRA_99** (Solids only)

SH1_SHELL_EXTRA_1 to **SH9_SHELL_EXTRA_99** (Shell and thick shells only)

Further derived values

SR_STRAIN_RATE (Solids only)

RV_RELATIVE_VOLUME ***VOLUME*** (Solids and thick shells only)

Averaged FORCE and MOMENT resultants in shells available at nodes

Thin shell force and moment resultants (see FX_ etc Explanation of Shell Force and Moment Resultants for an explanation of these)

FX_NORMAL_FORCE **MX_BENDING_MOMENT** **QXZ_SHEAR_FORCE**

FY_NORMAL_FORCE **MY_BENDING_MOMENT** **QYZ_SHEAR_FORCE**

FXY_SHEAR_FORCE **MXY_BENDING_MOMENT**

And the stresses derived by D3PLOT from these

XA_AXIAL_ONLY	**XB_BENDING_ONLY**	**XO_OUTER_FIBRE**
YA_AXIAL_ONLY	**YB_BENDING_ONLY**	**YO_OUTER_FIBRE**
XYS_SHEAR_ONLY		**XYO_OUTER_FIBRE**

Averaged thin shell miscellaneous components available at nodes

INTERNAL_ENERGY_DENSITY

AS_AREA_OF_SHELL **THICKNESS**

Averaged contact segment STRESS components available at nodes

Contact "stresses" in segments

CN_CONTACT_NORMAL	**CT_CONTACT_TANGENTIAL**
COX_CONTACT_X	**COY_CONTACT_Y**

16.7.4. Geometric Data Components Available for Output at Nodes

Geometric data components available for output at nodes

These may be listed using **WRITE** , but not plotted in any way.

Topological data:

EN_ELEMENTS_AT_NODE	(Lists elements attached to node)

16.8. Solid Element Data Components

16.8.1. The Results Available for Solid Elements

The results available for solid elements

Solid elements write the results to the `.PTF` file. The following tables show the raw data components, and those derived by D3PLOT.

Symmetric stress tensor:

X_DIRECT_STRESS	XY_SHEAR_STRESS
Y_DIRECT_STRESS	YZ_SHEAR_STRESS
Z_DIRECT_STRESS	ZX_SHEAR_STRESS

$$\begin{bmatrix} \sigma_{xx} & & \\ \tau_{yx} & \sigma_{yy} & \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$

This is written in the global cartesian coordinate system. It is always output, regardless of any of the switchable settings on the `*DATABASE_EXTENT_BINARY` control card. (See Controllable contents of the complete state file)

The stress components that can be derived by D3PLOT (Manipulations of Stress Tensor Components) are:

MAX_PRINC_STRESS	*MAX_DEV_PRINC_STRESS*	*VON_MISES_STRESS*
MID_PRINC_STRESS	*MID_DEV_PRINC_STRESS*	**SIGNED_VON_MISES_STRESS**
MIN_PRINC_STRESS	*MIN_DEV_PRINC_STRESS*	*MAX_SHEAR_STRESS*
YIELD_UTILISATION_FACTOR	*PRESSURE*	*LODE_PARAMETER*
YIELD_UTILISATION_PERCENTAGE	*TRIAXIALITY*	*LODE_ANGLE*
		LODE_PARAMETER_ALT

In addition D3PLOT will calculate the 2D in-plane maximum and minimum principal stresses and maximum shear stress:

S2MAX_2D_PRINC_STRESS	S2MIN_2D_PRINC_STRESS	S2MAX_2D_SHEAR_STRESS

These "2D" values are calculated by the following process:

- Rotate the global stress tensor to the element local axis system.
- Using ***only*** the local **Sxx** , **Syy** and **Txy** terms calculate the 2D max and min principal stresses

This means that the local **Szz, Tyz** and **Tzx** terms are implicitly treated as being zero and the element is treated as being plane stress, which is really a nonsense for a solid element in which a full three-dimensional stress state is achieved. These components are calculated for solid elements to provide consistency with thin and thick shell output, but the resulting values are of limited usefulness - use with care!

Effective plastic strain $[\epsilon_p]$

PLASTIC_STRAIN

The effective plastic strain is always output, as above. It has no intrinsic direction. (see A further explanation of strain components for more information about strains).

Extra variables for solids

If **NEIPH** has been defined on the ***DATABASE_EXTENT_BINARY** control card then **<neiph>** "extra" variables will be written. D3PLOT will accept any number of these, but will only process the first 99. These have the names:

SO1_SOLID_EXTRA_1 to **SO9_SOLID_EXTRA_99**

They are treated as separate scalar values of unknown type which may be contoured and written out, but not processed in any way.

Directional strain tensor for solids

If **STRFLG** has been set on the ***DATABASE_EXTENT_BINARY** control card the symmetric strain tensor for solids will be written out. This is always oriented in the global cartesian coordinate system. (see A further explanation of strain components for more information about strains)

$$\begin{matrix} [\epsilon_{xx} & &] \\ [\epsilon_{yx} & \epsilon_{yy} &] \\ [\epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz}] \end{matrix}$$

SX_DIRECT_STRAIN **SXY_SHEAR_STRAIN**

SY_DIRECT_STRAIN **SYZ_SHEAR_STRAIN**

SZ_DIRECT_STRAIN **SZX_SHEAR_STRAIN**

And the strains derived from these by D3PLOT (see Manipulations of Strain Tensor Components):

SMAX_PRINC_STRAIN ***SVON_MISES_STRAIN*** ***SAV_AVERAGE_STRAIN***

SMID_PRINC_STRAIN ***SMAX_SHEAR_STRAIN***

SMIN_PRINC_STRAIN ***PEMAG_PLAST_STRN_MAG***

The following components are the shear strain components multiplied by a factor of 2. See Manipulations of Strain Tensor Components for more details.

GXY_GAMMA_XY_STRAIN **GYZ_GAMMA_YZ_STRAIN** **GZX_GAMMA_ZX_STRAIN**

In addition D3PLOT will calculate the 2D in-plane maximum and minimum principal strains and maximum shear strain:

E2MAX_2D_PRINC_STRAIN **E2MIN_2D_PRINC_STRAIN** **E2MAX_2D_SHEAR_STRAIN**

These "2D" values are calculated by the following process:

- Rotate the global strain tensor to the element local axis system.
- Using ***only*** the local `Exx` , `Eyy` and `Exy` terms calculate the 2D max and min principal strains

This means that the local `Ezz` , `Eyz` and `Ezx` terms are all implicitly treated as being zero and the element is treated as being plane strain. Since solid elements develop a full 3D stress and strain state these are not really useful data components, and are computed only for consistency with thin and thick shell output - use with care!

D3PLOT will calculate the ratio of E2MIN_2D_PRINC_STRAIN / E2MAX_2D_PRINC_STRAIN which can be used to give an indication of the stress state:

E2D_PRINC_STRAIN_RATIO

A problem arises when trying to distinguish between biaxial tension and biaxial compression as in both cases they return a +ve value. In fracture predictions it is important to distinguish between the two so, whilst it is possbile that a compressive state could give a shear fracture, the most likely outcome would be buckling. Therefore, when both E2MAX_2D_PRINC_STRAIN and E2MIN_2D_PRINC_STRAIN are negative (or 0) a value of 1.1 is returned to indicate a compressive state.

The ratio is capped to -100 to avoid large values if E2MAX_2D_PRINC_STRAIN is relatively small compared to E2MIN_2D_PRINC_STRAIN

Internal Energy Density for solids

For solid elements, D3PLOT may derive the internal energy density for solids in the elastic regime. This is not computed by default, users wishing to use it should contact Oasys Ltd first for advice.

Components derived geometrically by D3PLOT

Strain rate is calculated directly from the translational velocity gradients and nodal displacements, giving an instantaneous result at a given time:

SR_STRAIN_RATE

Volume and relative volume ($\mathrm{Vol}_{\mathrm{CURRENT}}$ / $\mathrm{Vol}_{\mathrm{Original}}$) are calculated from nodal coordinates:

RV_RELATIVE_VOLUME ***VOLUME***

Geometric components

These components are extracted from the topology, and can be output in **WRITE** :

MN_MATERIAL_NUMBER **LN_LIST_OF_NODES** **FE_FACING_ELEMENTS**

16.8.2. Averaged Nodal Components for Solids

Averaged nodal components for solids

All the nodal results from the **.PTF** file listed in Section 9.6.1 are available for contouring on solid elements (in 2D/3D plotting mode). They can also be extracted as averaged element data for use in **WRITE** and **XY_PLOT** .

Nodal contact force components from the **.CTF** file cannot be processed on solid elements.

16.8.3. Transforming Directional Solid Results to the Element Local System

Transforming directional solid results to the element local system

It is possible to transform stress and strain tensor results from the global to the local element coordinate system using the **FRAME_OF_REFERENCE** options.

The local element axes, **[X', Y', Z']** for this purpose, are calculated as follows. If we adopt the notation that the vector from node 1 to node 2 is `N1N2` :

`X'`	=	`N1N2`	approximately: the true `X'` vector is recomputed below
`Z'`	=	`N1N3 x N2N4`	(Where `x` is a vector cross-product)
`Y'`	=	`Z' x X'`	
`X'`	=	`Y' x Z'`	This final transformation is required to correct for any warping

The formulae above are simplified for clarity. In 8 noded hexahedra the average of the bottom (N1N2N3N4) and top (N5N6N7N8) faces is used to determine a "middle" face; and for 6 noded wedges a similar averaging process is used. For tetrahedra Z' is obtained from N1N2 x N1N3.

16.8.4. Solid Element Results from Thermal-Only (TOPAZ3D) Analyses

Solid element results from thermal-only (TOPAZ3D) analyses

When a thermal-only analysis is run solid elements only are used. However these do not write any results: the only output from such analyses is temperatures and flux vectors (in place of velocities) at nodes.

Thermal loading for structural models: the *LOAD_THERMAL_TOPAZ keyword

It is possible to apply thermal loading to a structural model in the form of temperatures at nodes, varying over time, by using a "topaz" input file combined with the *LOAD_THERMAL_TOPAZ keyword. However this has to be a "pure" thermal file, formatted exactly as if it came from an old (1980s or 1990s) version of Topaz3d, and this is not the same as the output file that contemporary LS-DYNA generates from a thermal only analysis.

D3PLOT has an undocumented capability to generate "pure" Topaz files from contemporary "thermal only" output - please contact Oasys Ltd if you need to use this feature.

16.8.5. Solid Element Results from Combined Thermal and Structural Analyses

Solid element results from combined thermal and structural analyses

When a combined thermal/structural analysis is run the results for solids are exactly the same as those in a purely structural analysis.

16.8.6. Solid Element Results from an Implicit NIKE3D Analysis

Solid element results from an implicit NIKE3D analysis

D3PLOT provides limited support for standalone NIKE3D (`N3PLOT`) files. Solid elements in these files report results at 8 integration points, and D3PLOT will process this, but this capability is still under development. Users wishing to use NIKE3D, either standalone or as embedded in LS-DYNA, should contact Oasys Ltd first for advice.

16.8.7. User-Defined Solid Data Components

User-defined Solid data components

D3PLOT 9.3 permits you to create an unlimited number of user-defined data components for solids. These may be either scalar or tensor:

- Scalar components are just treated as numerical values with no known properties or orientation.
- Tensor components are assumed to be in the global system when created, and are subject to Frame of Reference transformation as described in Transforming directional solid results to the element local system in exactly the same way as stresses and strains read from the database. The same derived components (principal, max shear, von Mises, etc) are also available.

16.8.8. Deletion Time

Deletion time

The deletion time of solid elements may now be plotted using this data component. Solid elements that aren't deleted return a value of 0.0 for the deletion time.

This value is invariant over time. The same deletion time is returned irrespective of the current animation state.

Deletion time may also be obtained for shell elements.

16.9. Thin Shell Element Results

16.9.1. Effect of Shell Shape and Formulation on Output

Effect of shell shape and formulation on output

Thin shells in LS-DYNA may be quadrilateral or triangular; they may have a single integration point on plan, or be fully integrated; they may have from 1 to <n> integration points through their thickness; they may be linear (3 or 4 noded) or parabolic (6 or 8 noded).

Regardless of any of these permutations:

- ***On plan:*** ***If `<maxint>` is POSITIVE then shells only report results at their centre. For fully integrated shells this will be the averaged value of the in-plane integration points.***

  If `<maxint>` is NEGATIVE then shells will report results at 2 x 2 in-plane integration points.
- ***In elevation:*** ***Shells only report results at `<maxint>` points through their thickness.***

(**`<maxint>`** is on the **`*DATABASE_EXTENT_BINARY`** card, and is explained in Controllable contents of the complete state file)

Sounds simple? Well it isn't! There is lots of scope for misunderstanding thin shell results, and if you have doubts please read this section. If you are still confused after that please contact Oasys Ltd for help and advice.

16.9.2. Description of Shell Output

Description of shell output

Thin shells write out a large number of data values, and it is important that you understand exactly what they are. The greatest sources of confusion are the coordinate systems used and the relationship between integration points and data.

16.9.2.1. Thin Shell Coordinate Systems

Thin shell coordinate systems

Global Cartesian and element local.

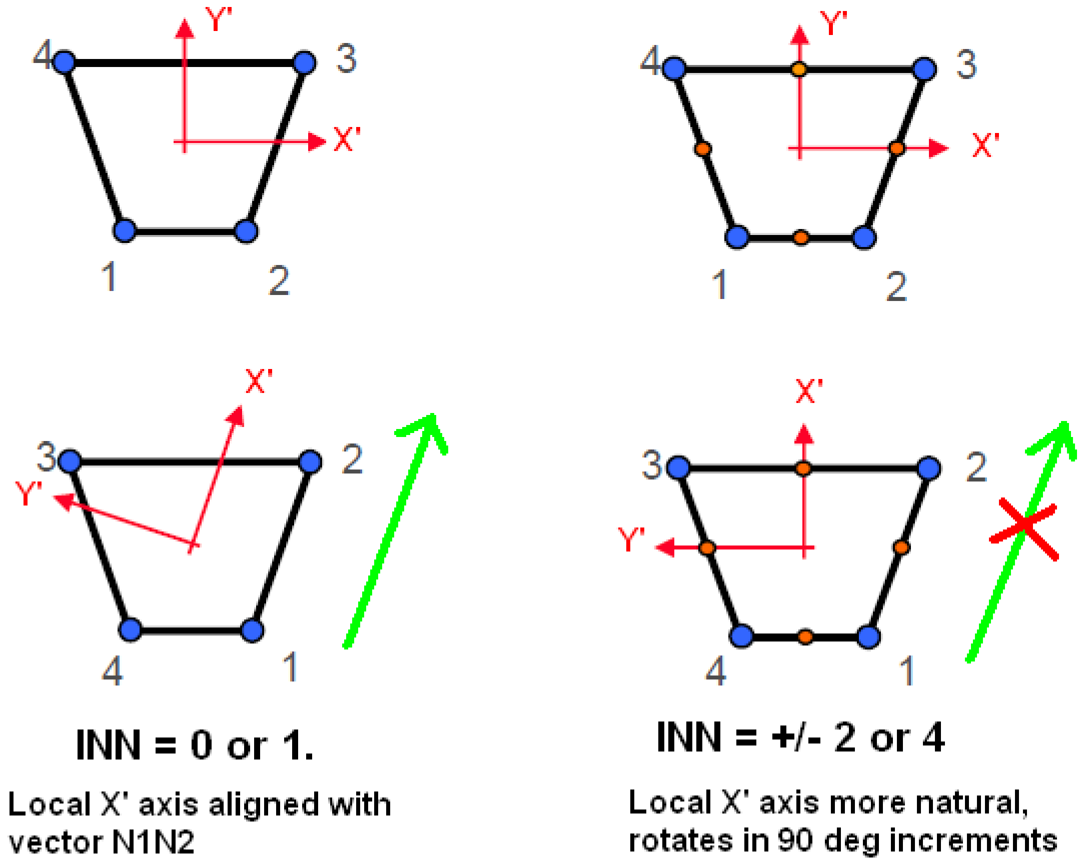
By default thin shell results are written in the following coordinate systems:

- Tensor components such as stress and strain are written in the global system
- Force and moment resultants are written in the element local system
- Other results, eg effective plastic strain, thickness, etc have no intrinsic coordinate system.

It is possible to write tensor results in other systems by using special flags on some material cards. If you do this be aware that there is no flag in the results database to indicate the coordinate sytem used, so D3PLOT will not "know about" this and will assume that these results are in the global system.

Effect of INN flag on *CONTROL_ACCURACY card on local coordinate system axes

By default element local axes in LS-DYNA align their local X' axis with the vector N1N2. However if the element is poorly shaped, or becomes heavily deformed during the analysis, this can degrade the accuracy of the stress calculation as the original stress directions become misaligned with the updated axes, so an improved calculation method is provided via flag **`INN`** on **`*CONTROL_ACCURACY`** , although this costs more cpu time. The diagram below shows how the local axis system ceases to be so dependent on the N1N2 vector.



The **INN** value is reported in the d3plot (.ptf) database file for versions of LS-DYNA from approximately 2012 onwards (exact version not known at time of writing). If it is present and set to a non-default value D3PLOT will adjust its local axis calculation for shell elements to use the correct axis system. This will affect the following:

- Calculation of cut-section forces through shell elements, which use the shell force and moment resultants
- Conversion of tensor-based results from global to local axes ("Local" frame of reference)
- Conversion of tensor-based results into composite ply local axes.
- Display of local element triads.

The local element axes, **[X',Y',Z']** , are calculated as follows.

Default case: INN = 0 or 1 on *CONTROL_ACCURACY

If we adopt the notation that the vector from node 1 to node 2 is **N1N2** :

X'	=	**N1N2**	approximately: the final **X'** vector recomputed below.
Z'	=	**N1N3 x N2N4**	(Where **x** is a vector cross-product)
Y'	=	**Z' x X'**	

`X'` = `Y'` `x` `Z'` This final transformation is required to correct for any warping

Some components are written in the global Cartesian system, and D3Plot can transform these results to the local element system if required with the **FRAME_OF_REFERENCE** options. Examples are the stress and strain tensor values.

Other components are written in the element local system. Examples are the force and moment resultants.

Invariant case: INN = +/-2 or +/-4 on *CONTROL_ACCURACY

A more complex vector calculation is performed to obtain more natural coordinates, as shown in the image above.

We will revisit this topic in Summary of coordinate systems and default locations of thin shell results.

16.9.2.2. Thin Shell Integration Points

Thin shell integration points

From 1 to n through the thickness.

The majority of shell element formulations in LS-DYNA have a single integration point on plan, with from 1 to <n> integration points through their thickness. The "fully integrated" formulations have 2 x 2 integration points on plan, and again from 1 to <n> points through the thickness. For more information about shell formulations refer to the LS-DYNA theory manual.

If **`MAXINT`** is a positive value then LS-DYNA will write results at the element centre as viewed on plane. For fully integrated formulations this will be the averaged value of the in-plane integration points.

If **`MAXINT`** is a negative value then LS-DYNA will write results at 2 x 2 integration points as viewed on plane. This is true for all element formulations regardless of whether they are fully integrated or not. If a ZTF file has been written from PRIMER, then D3PLOT will be able to determine which integration point results are valid for each Shell element and disregard them if they are not relevant. Without the ZTF file D3PLOT will not have the information required to determine this and will only read data from the first on plan integration point.

For the purposes of this discussion we will consider the case where **`MAXINT`** is positive since we can then ignore the number of integration points on plan. What we are concerned with here is the number through the thickness, (as viewed in elevation) and we must consider two cases:

Case 1: `MAXINT` on the `*DATABASE_EXTENT_BINARY` card is set to the default of three.

In this situation, ***regardless of the number of integration points through the thickness used in the shell element formulation(s***), stress tensor and plastic strain output for all shell elements is written at three "surfaces":

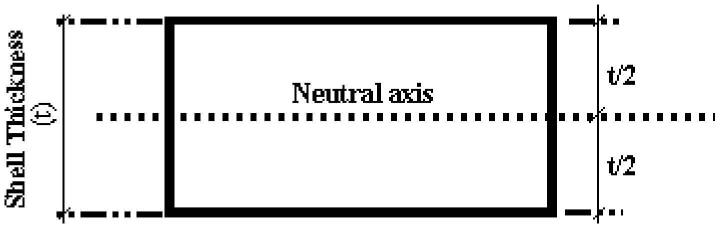
"Top"	:	The outermost integration point
"Middle"	:	The neutral axis
"Bottom"	:	The innermost integration point

The outer integration point is located on the +ve local Z' of the neutral axis, the inner on the -ve Z' side. (Membrane elements with a single integration point write the same value to all three surfaces.)

The important thing to note here is that, unlike linear-elastic codes, LS-DYNA reports "top and bottom surface" stresses at the outer and inner integration points, and ***not at the element outer fibres.***

Assuming the default Gaussian integration scheme is used the location of the outermost integration points is given in the following table as a function of shell thickness/2 (`t/2`) for the range of 1 to 6 integration points through the thickness.

No of Points	**Distance from neutral axis as a proportion of** `t/2` **(Gaussian integration)**
1	0.0 (membrane)
2	0.577
3	0.775
4	0.861
5	0.906
6	0.932



Note:

Prior to LS-960	If the element formulation uses >5 integration points through its thickness then trapezoidal integration is used - even if Gaussian is specified..
From LS-960 onwards	The user manual claims that Gaussian integration is used for any number of points if specified.

It is possible to specify "user-defined" integration rules, which may locate integration points at different points through the thickness. For example Gauss-Lobatto quadrature will locate the outer and inner integration points on their respective outer fibres, but there is a speed penalty to pay for this.

In addition, in many analyses the section will be largely plastic, and the exact values of stress will be somewhat academic. Generally highly nonlinear problems are more concerned with plasticity and consequent energy absorption. However if you are in, or close to, the linear elastic region, and/or exact stress levels are important, you should consider using more integration points in order to extract results more precisely.

Case 2: "Layer" output with `MAXINT` set to some value other than the default of 3

In this situation LS-DYNA writes out the stress tensor and plastic strain values for the first `<maxint>` integration points in the shell, starting at the innermost (bottom) one. You would generally only use this for composites analysis where results in all layers are significant.

(1) LS-DYNA no longer calculates the "middle" neutral axis values for you.

If you have an odd number of integration points D3Plot assumes that the middle one is the neutral axis value, if you have an even number of integration points D3Plot averages the central pair to produce an approximate "middle" value.

(2) There is no check that `MAXINT` is equal to #integration points in a shell.

If `MAXINT` is less than the number of integration points in a shell you will be missing results; if greater the extra results will probably be junk. The true number of integration points in a shell is not written to the `.PTF` file, so D3PLOT cannot check this for you.

(3) There is a special case when the #integration points in a shell = 5.

If, and only if, #integration points = 5 then results are written out in the integration point order: #3, #1, #2, #4, #5. There is no obvious reason for LS-DYNA doing this, and it is not documented anywhere, but a simple shell bending test model will reveal it to be true.

In this situation D3Plot will still permit you to select "top", "middle" and "bottom" surfaces as before (with the "middle" surface being defined as in (1) or (3) above). But it will also permit you to define a Layer number instead in the range 1 to `<maxint>` . An example of this is shown in SURFACE / INT Point .

If you choose this output option it is your responsibility to ensure that you interpret your results correctly.

Case 3: "Composite Ply" output with `MAXINT` set to some value other than the default of 3

From Version 13 onwards if the model contains composite plys then D3Plot will permit users to plot data on a surface composed of plys (see SURFACE with composite plys). This requires composites to be set-up in PRIMER using the Composites tool, or equivalently *SHELL_COMPOSITE_LONG cards, and a `.ztf` file. (Composite plys created using a *PART_COMPOSITE card are not available in this feature.) To plot the data a surface composed of plys D3Plot extracts the data for each shell at the integration point the composite ply corresponds to.

To summarise the order of LS-DYNA output for Thin Shells is:

MAXINT value	**Surface output order**
3	Always MIDDLE, BOTTOM, TOP
Anything else	If #integration points = 3, then MIDDLE, BOTTOM, TOP If #integration points = 5, then MIDDLE, BOTTOM, next BOTTOM, next Anything else: in order from BOTTOM to TOP

If a **`.ZTF`** file is present, D3Plot will know how many integration points a shell has and will therefore be able to determine the correct value to read for a selected surface, based on the rules above.

From v11.0 onwards Layer 1->Layer n is always Bottom->Top (so long as a **`.ZTF`** file is present). Prior to this the layers were in the order of the integration points output by LS-DYNA, e.g. for < **`maxint`** >=3 Layer 1 was the MIDDLE surface, Layer 2 was the BOTTOM surface and Layer 3 was the TOP surface.

> If a **`.ZTF`** file is not present D3PLOT has no means of knowing how many integration points a given element has actually specified on its section definition; nor whether it uses Gaussian, Lobatto or user-defined rules.
>
> ***It is up to you to interpret your results correctly.***

16.9.2.3. Cases Where Stress and Strain Tensor Output are in the Local Coordinate System

Cases where stress and strain tensor output are in the local coordinate system

Flag **CMPFLG** on the ***DATABASE_EXTENT_BINARY** card can be set to cause composite material stress and strain tensor output to be in the element (or material) local coordinate system. There is no way to detect this from the the **.PTF** file so, if you use this option, D3PLOT will assume that your results are still in the global system. Therefore it will be ***your responsibility to ensure that you interpret your results correctly.***

16.9.3. The Results Available for Thin Shell Elements

The results available for thin shell elements

Thin shell elements write the results to the **.PTF** file. The following tables show the raw data components, and those derived by D3PLOT.

Symmetric stress tensor:

X_DIRECT_STRESS	XY_SHEAR_STRESS	
Y_DIRECT_STRESS	YZ_SHEAR_STRESS	
Z_DIRECT_STRESS	ZX_SHEAR_STRESS	

$$\begin{bmatrix} \sigma_{xx} & & \\ \tau_{yx} & \sigma_{yy} & \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$

This is assumed to be written in the global Cartesian coordinate system at **<maxint>** surfaces. Its output is on by default, but may be switched off using the **SIGFLG** parameter on the ***DATABASE_EXTENT_BINARY** control card. (See Controllable contents of the complete state file)

The stress components that can be derived by D3PLOT (Manipulations of Stress Tensor Components) are:

MAX_PRINC_STRESS	*MAX_DEV_PRINC_STRESS*	*VON_MISES_STRESS*	*PRESSURE*
MID_PRINC_STRESS	*MID_DEV_PRINC_STRESS*	SIGNED_VON_MISES_STRESS	*TRIAXIALITY*
MIN_PRINC_STRESS	*MIN_DEV_PRINC_STRESS*	*MAX_SHEAR_STRESS*	*LODE_ANGLE*
YIELD_UTILISATION_FACTOR		*LODE_PARAMETER*	*LODE_PARAMETER_ALT*

YIELD_UTILISATION_PERCENTAGE ***TC_SHELL_TENS_COMP***

In addition D3PLOT will calculate the 2D in-plane maximum and minimum principal stresses and maximum shear stress:

S2MAX_2D_PRINC_STRESS **S2MIN_2D_PRINC_STRESS** **S2MAX_2D_SHEAR_STRESS**

These "2D" values are calculated by the following process:

- Rotate the global stress tensor to the element local axis system.
- Using ***only*** the local **Sxx** , **Syy** and **Txy** terms calculate the 2D max and min principal stresses

This means that the local **Tyz** and **Tzx** terms are implicitly treated as being zero and the element is treated as being plane stress, which may be more or less true in the middle of a panel but is unlikely to be the case at the edges of a panel and at connections. Therefore please use these terms with care.

Effective plastic strain $[\epsilon_p]$

PLASTIC_STRAIN

The effective plastic strain is written at the same **<maxint>** surfaces as the stress tensor. It has no intrinsic direction. By default it is switched **on** , but may be turned **off** using the **EPSFLG** parameter on the ***DATABASE_EXTENT_BINARY** control card. (see A further explanation of strain components for more information about strains)

Strain rate for thin shells

SR_STRAIN_RATE

For shell elements where bending takes place knowledge of nodal rotations would be required, and this information is not available. Therefore the strain rate in shells is approximated by:

$(\varepsilon p1 - \varepsilon p0) / (t1 - t0)$ where

$\varepsilon p1$ is the effective plastic strain at this state, time t1

$\varepsilon p0$ is the effective plastic strain at the previous state, time t0

This value is a poor approximation because effective plastic strain will only ever increase, so the result will always be zero or positive, and moreover it does not consider any elastic strains. Nevertheless in a loading regime that is mostly uniaxial it gives a reasonable result.

Extra variables for thin shells

If **NEIPS** has been defined on the ***DATABASE_EXTENT_BINARY** control card then **<neips>** "extra" variables will be written at each of **<maxint>** surfaces (as for the stress tensor above). D3PLOT will accept any number of these, but will only process the first 99. These have the names:

SH1_SHELL_EXTRA_1 to **SH9_SHELL_EXTRA_99**

They are treated as separate scalar values of unknown type which may be contoured and written out, but not processed in any way.

Directional strain tensor for thin shells

If **STRFLG** has been set on the ***DATABASE_EXTENT_BINARY** control card the symmetric strain tensor for thin shells will be written out at the innermost and outermost integration points only, regardless of the value of **<maxint>** . This is assumed to be oriented in the global Cartesian coordinate system. (see A further explanation of strain components for more information about strains)

$$\begin{bmatrix} \epsilon_{xx} & & \\ \epsilon_{yx} & \epsilon_{yy} & \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix}$$

SX_DIRECT_STRAIN	**SXY_SHEAR_STRAIN**
SY_DIRECT_STRAIN	**SYZ_SHEAR_STRAIN**
SZ_DIRECT_STRAIN	**SZX_SHEAR_STRAIN**

And the strains derived from these by D3PLOT (Manipulations of Strain Tensor Components):

SMAX_PRINC_STRAIN	***SVON_MISES_STRAIN***	***SAV_AVERAGE_STRAIN***
SMID_PRINC_STRAIN	***SMAX_SHEAR_STRAIN***	

SMIN_PRINC_STRAIN ***PEMAG_PLAST_STRN_MAG***

For shells, the engineering strains are derivied (Manipulations of Strain Tensor Components):

ENG_MAJOR_STRAIN **ENG_MINOR_STRAIN** **ENG_THICKNESS_STRAIN**

The following components are the shear strain components multiplied by a factor of 2. See Manipulations of Strain Tensor Components for more details.

GXY_GAMMA_XY_STRAIN **GYZ_GAMMA_YZ_STRAIN** **GZX_GAMMA_ZX_STRAIN**

In addition D3PLOT will calculate the 2D in-plane maximum and minimum principal strains and maximum shear strain:

E2MAX_2D_PRINC_STRAIN **E2MIN_2D_PRINC_STRAIN** **E2MAX_2D_PRINC_STRAIN**

These "2D" values are calculated by the following process:

- Rotate the global strain tensor to the element local axis system.
- Using ***only*** the local `Exx` , `Eyy` and `Exy` terms calculate the 2D max and min principal strains

This means that the local `Ezz` , `Eyz` and `Ezx` terms are all implicitly treated as being zero and the element is treated as being plane strain. Bearing in mind that LS-DYNA populates all 6 terms of the strain tensor (local `Ezz` being finite due to the need to conserve volume) this is a gross simplification, and these terms should be used with care.

D3PLOT will calculate the ratio of E2MIN_2D_PRINC_STRAIN / E2MAX_2D_PRINC_STRAIN which can be used to give an indication of the stress state:

E2D_PRINC_STRAIN_RATIO

A problem arises when trying to distinguish between biaxial tension and biaxial compression as in both cases they return a +ve value. In fracture predictions it is important to distinguish between the two so, whilst it is possible that a compressive state could give a shear fracture, the most likely outcome would be buckling. Therefore, when both E2MAX_2D_PRINC_STRAIN and E2MIN_2D_PRINC_STRAIN are negative (or 0) a value of 1.1 is returned to indicate a compressive state.

The ratio is capped to -100 to avoid large values if E2MAX_2D_PRINC_STRAIN is relatively small compared to E2MIN_2D_PRINC_STRAIN.

Shell force and moment resultants

Shell force and moment resultants are written as **`<Force/unit width>`** and **`<moment/unit width>`** in the element local coordinate system. By default these are **on** , but they can be switched **off** with the **`RLTFLG`** flag on the **`*DATABASE_EXTENT_ BINARY`** control card. (These components are explained in FX_ etc Explanation of Shell Force and Moment Resultants .)

FX_NORMAL_FORCE	**MX_BENDING_MOMENT**	**QXZ_SHEAR_FORCE**
FY_NORMAL_FORCE	**MY_BENDING_MOMENT**	**QYZ_SHEAR_FORCE**
FXY_SHEAR_FORCE	**MXY_BENDING_MOMENT**	

Stresses derived by D3PLOT from the force & moment resultants. (These components are explained in XA_ etc Stresses in Thin Shells Derived from Force and Moment Resultants .)

XA_AXIAL_ONLY	***XB_BENDING_ONLY***	***XO_OUTER_FIBRE***
YA_AXIAL_ONLY	***YB_BENDING_ONLY***	***YO_OUTER_FIBRE***
XYS_SHEAR_ONLY		***XYO_OUTER_FIBRE***

Thickness and Internal energy density components

By default these components are written. They can be turned **off** with the **`ENGFLG`** flag on the **`*DATABASE_EXTENT_BINARY`** control card.

INTERNAL_ENERGY_DENSITY

THICKNESS

Components derived geometrically by D3PLOT

Shell area is calculated from the nodal coordinates:

AS_AREA_OF_SHELL *ON_OUTWARD_NORMAL*

Geometric components

These components are extracted from the topology, and can be output in **WRITE** :

MN_MATERIAL_NUMBER **LN_LIST_OF_NODES**

16.9.4. Averaged Nodal Data Components for Thin Shells

Averaged nodal data components for thin shells

All the nodal results from the `.PTF` file listed in Section 13.6.1 are available for contouring on the shell elements (in 2D/3D plotting mode). They can also be extracted as averaged element data for use in **WRITE** and **XY_PLOT** .

Nodal contact force components from the `.CTF` file **cannot** be processed on thin shell elements.

16.9.5. Out of Plane (Z') Stress Tensor Components

Out of plane (Z') stress tensor components

Thin shells in LS-DYNA do not calculate the element through-thickness ($\sigma_{Z'}$) direct stress which will always be zero, although the through thickness shear terms ($\tau_{YZ'}$ and $\tau_{XZ'}$) are calculated. Thus five of the six terms of the local stress tensor are used, giving:

$$\begin{bmatrix} \sigma'_{xx} & & \\ \tau'_{yx} & \sigma'_{yy} & \\ \tau'_{zx} & \tau'_{zy} & 0 \end{bmatrix}$$

This means that principal stresses will not necessarily lie in the plane of the shell.

16.9.6. Out of Plane (Z') Strain Tensor Components

Out of plane (Z') strain tensor components

Thin shells do not integrate the through thickness strain ($\varepsilon_{ZZ'}$), but this value may nevertheless be reported as non-zero. This is because the shell thickness may change, leading to a strain: its stiffness in this direction is based on volume preservation, but there is no stress associated with this.

Therefore principal strains also may not be confined to the plane of the element.

16.9.7. FX_ etc Explanation of Shell Force and Moment Resultants

FX_ etc Explanation of shell force and moment resultants

These are integrals, computed by LS-DYNA, of the stresses in the shell local system to create force and moment values per unit width in the element local system.

The sign convention is taken from "Theory of Elastic Stability" Timoshenko and Gere, and the equations to derive the resultants from the local stress values are given below. (**t** is the shell thickness.)

$$F_X = \int_{-t/2}^{+t/2} \sigma'_X \cdot dt \qquad F_Y = \int_{-t/2}^{+t/2} \sigma'_Y \cdot dt \qquad F_{XY} = \int_{-t/2}^{+t/2} \tau'_{XY} \cdot dt$$

$$M_X = \int_{-t/2}^{+t/2} t \cdot \sigma'_X \cdot dt \qquad M_Y = \int_{-t/2}^{+t/2} t \cdot \sigma'_Y \cdot dt \qquad M_{XY} = \int_{-t/2}^{+t/2} t \cdot \tau'_{XY} \cdot dt$$

$$Q_{XZ} = \int_{-t/2}^{+t/2} \tau'_{XZ} \cdot dt \qquad Q_{YZ} = \int_{-t/2}^{+t/2} \tau'_{YZ} \cdot dt$$

These are the forces and moments per unit width integrated over the element thickness. They are written in the element local axis system.

16.9.8. XA_ etc Stresses in Thin Shells Derived from Force and Moment Resultants

XA_ etc Stresses in thin shells derived from force and moment resultants

The force/moment resultants are converted to local axial and bending stresses as follows:

Axial stress	=	`F/t`
Bending stress	=	`6M/t` ^{2}
Outer fibre stress	=	`F/t +/- 6M/t` ^{2}

Where:

`F` = force / unit width

`M` = moment / unit width

`t` = section thickness

Note that the bending component here assumes a ***linear elastic stress*** distribution (ie stress = `My/I`) so "bending" and "outer fibre" stresses will be ***incorrect*** in a plastic section.

16.9.9. Summary of Coordinate Systems and Default Locations of Thin Shell Results

Summary of coordinate systems and default locations of thin shell results

This table shows the coordinate system and location of each category of data written by LS-DYNA, and also the same information for results derived from these by D3Plot. It assumes the default of `MAXINT` = 3.

TYPE OF DATA	COORDINATE SYSTEM	ELEMENT LOCATION
Basic global stress tensor	Global Cartesian axes (can transform)	Inner integration point Neutral axis Outer integration point
Stresses derived from global tensor	n/a	Inner integration point Neutral axis Outer integration point
Plastic strain	n/a	Inner integration point Neutral axis Outer integration point
Force/moment resultants	Element local axes	Whole element

Stresses derived from force/moment resultants	Element local axes	Outer fibre Neutral axis
Directional strain tensor	Global Cartesian axes (can transform)	Inner integration point Outer integration point
Extra data at integration points	Undefined	Inner integration point Neutral axis Outer integration point
Additional components	n/a	Whole element

Coordinate system and default locations for thin shell element data

Notes on this table:

- The entry "n/a" in the coordinate system column means that the data does not have a fixed direction. For example **VON_MISES** or **PRINCIPAL** stresses. The words "can transform" means that D3Plot can transform results to other co-ordinate systems (element local, cylindrical, user-defined and ply local).

- The entry "whole element" in the element location column means that the data is for the element as a whole. For example **FXX_AXIAL_FORCE** or **THICKNESS** components.

- The "outer fibre" stresses calculated from force/moment resultants assume a linear elastic stress distribution the element. This may not always be the case.

16.9.10. User-Defined Shell Components

User-defined shell components

D3PLOT 9.3 permits you to create an unlimited number of user-defined data components for shells. These may be either scalar or tensor:

- Scalar components are just treated as numerical values with no known properties or orientation.
- Tensor components are assumed to be in the global system when created, and are subject to Frame of Reference transformation in exactly the same way as stresses and strains read from the database. The calculation of the element local system is explained in Thin shell coordinate systems above. The same derived components (principal, max shear, von Mises, etc) are also available.

16.10. Thick Shell Element Results

Thick shell element results

Thick shells in LS-DYNA look like bricks, but the default formulation for thick shells is effectively an extruded thin shell element. They use very similar integration schemes and, like thin shells, do not integrate through thickness (Z') stress. In this case, through thickness strain is computed from volume conservation.

Also available are fully integrated formulations (ELFORM=3 onwards) that are layered brick elements which ***do*** compute through thickness stresses.

16.10.1. Effect of Shell Shape and Formulation on Output

Effect of shell shape and formulation on output

Thick shells in LS-DYNA may be hexahedra or extruded triangles. They may have a single integration point on plan, or be fully integrated. They may have from 1 to <n> integration points through their thickness.

However, like thin shells:

On plan:	***thick shells only report results at their centre;***
In elevation:	***thick shells only report results at `<maxint>` points through their thickness.***

(**`<maxint>`** is on the **`*DATABASE_EXTENT_BINARY`** card, and is explained in Controllable contents of the complete state file

In most respects thick shells are like thin shells, and you are encouraged to read Description of shell output which describes some of the pitfalls of using them. This will not be repeated here.

For thick shells, if <maxint> = 3, the order of output is middle, bottom, top. For any other value of <maxint> the order of output is always #1, #2, #3 ... #nip, i.e. from bottom to top.

NOTE: For thick shells, ELFORM=1, if <NIP> on the *SECTION card = 2 then LS-DYNA switches it to 3 integration points. Similarly, if <NIP> = 4 it is switched to 5.

To summarise the order of output for Thick Shells is:

MAXINT value	Surface output order
3	Always MIDDLE, BOTTOM, TOP

Anything else	In order from BOTTOM to TOP

If a ZTF file is present, D3PLOT will know how many integration points a shell has and will therefore be able to determine the correct value to read for a selected surface, based on the rules above.

16.10.2. The Results Available for Thick Shell Elements

The results available for thick shell elements

Thick shell elements write the results to the `.PTF` file. The following tables show the raw data components, and those derived by D3PLOT.

Symmetric stress tensor:

X_DIRECT_STRESS	XY_SHEAR_STRESS
Y_DIRECT_STRESS	YZ_SHEAR_STRESS
Z_DIRECT_STRESS	ZX_SHEAR_STRESS

$$\begin{bmatrix} \sigma_{xx} & & \\ \tau_{yx} & \sigma_{yy} & \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$

This is assumed to be written in the global cartesian coordinate system at `<maxint>` surfaces. Its output is **on** by default, but may be switched **off** using the `SIGFLG` parameter on the `*DATABASE_EXTENT_BINARY` control card. (See Controllable contents of the complete state file)

The stress components that can be derived by D3PLOT (Manipulations of Stress Tensor Components) are:

MAX_PRINC_STRESS	*MAX_DEV_PRINC_STRESS*	*VON_MISES_STRESS*	*PRESSURE*
MID_PRINC_STRESS	*MID_DEV_PRINC_STRESS*	SIGNED_VON_MISES_STRESS	*TRIAXIALITY*
MIN_PRINC_STRESS	*MIN_DEV_PRINC_STRESS*	*MAX_SHEAR_STRESS*	*LODE_ANGLE*
YIELD_UTILISATION_FACTOR	*YIELD_UTILISATION_PERCENTAGE*	*LODE_PARAMETER*	*LODE_PARAMETER_ALT*

In addition D3PLOT will calculate the 2D in-plane maximum and minimum principal stresses and maximum shear stress:

S2MAX_2D_PRINC_STRESS S2MIN_2D_PRINC_STRESS S2MAX_2D_SHEAR_STRESS

These "2D" values are calculated by the following process:

- Rotate the global stress tensor to the element local axis system.
- Using ***only*** the local **Sxx** , **Syy** and **Txy** terms calculate the 2D max and min principal stresses

This means that the local **Tyz** and **Tzx** terms are implicitly treated as being zero and the element is treated as being plane stress, which may be more or less true in the middle of a panel but is unlikely to be the case at the edges of a panel and at connections. Therefore please use these terms with care.

Effective plastic strain $[\epsilon_p]$

PLASTIC_STRAIN

The effective plastic strain is written at the same **<maxint>** surfaces as the stress tensor. It has no intrinsic direction. By default it is switched on, but may be turned off using the **EPSFLG** parameter on the ***DATABASE_EXTENT_BINARY** control card. (see A further explanation of strain components for more information about strains)

Strain rate for thin shells

SR_STRAIN_RATE

For shell elements where bending takes place knowledge of nodal rotations would be required, and this information is not available. Therefore the strain rate in shells is approximated by:

$(\varepsilon p1 - \varepsilon p0) / (t1 - t0)$ where

$\varepsilon p1$ is the effective plastic strain at this state, time t1

$\varepsilon p0$ is the effective plastic strain at the previous state, time t0

This value is a poor approximation because effective plastic strain will only ever increase, so the result will always be zero or positive, and moreover it does not consider any elastic strains. Nevertheless in a loading regime that is mostly uniaxial it gives a reasonable result.

Extra variables for thick shells

If **NEIPS** has been defined on the ***DATABASE_EXTENT_BINARY** control card then **<neips>** "extra" variables will be written at each of **<maxint>** surfaces (as for the stress tensor above). D3PLOT will accept any number of these, but will only process the first 99. These have the names:

SH1_SHELL_EXTRA_1 to **SH9_SHELL_EXTRA_99**

They are treated as separate scalar values of unknown type which may be contoured and written out, but not processed in any way.

Directional strain tensor for thick shells

If **STRFLG** has been set on the ***DATABASE_EXTENT_BINARY** control card the symmetric strain tensor for thick shells will be written out ***at the innermost and outermost integration points only, regardless of the value of*** *<maxint>* . This is assumed to be oriented in the global cartesian coordinate system. (see A further explanation of strain components for more information about strains)

$$\begin{bmatrix} \epsilon_{xx} & & \\ \epsilon_{yx} & \epsilon_{yy} & \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix}$$

SX_DIRECT_STRAIN **SXY_SHEAR_STRAIN**

SY_DIRECT_STRAIN **SYZ_SHEAR_STRAIN**

SZ_DIRECT_STRAIN **SZX_SHEAR_STRAIN**

And the strains derived from these by D3PLOT (Manipulations of Stress Tensor Components):

SMAX_PRINC_STRAIN ***SVON_MISES_STRAIN*** ***SAV_AVERAGE_STRAIN***

SMID_PRINC_STRAIN ***SMAX_SHEAR_STRAIN***

SMIN_PRINC_STRAIN ***PEMAG_PLAST_STRN_MAG***

The following components are the shear strain components multiplied by a factor of 2. See Manipulations of Stress Tensor Components for more details.

GXY_GAMMA_XY_STRAIN **GYZ_GAMMA_YZ_STRAIN** **GZX_GAMMA_ZX_STRAIN**

In addition D3PLOT will calculate the 2D in-plane maximum and minimum principal strains and maximum shear strain:

E2MAX_2D_PRINC_STRAIN **E2MIN_2D_PRINC_STRAIN** **E2MAX_2D_SHEAR_STRAIN**

These "2D" values are calculated by the following process:

- Rotate the global strain tensor to the element local axis system.
- Using ***only*** the local **`Exx`** , **`Eyy`** and **`Exy`** terms calculate the 2D max and min principal strains

This means that the local **`Ezz`** , **`Eyz`** and **`Ezx`** terms are all implicitly treated as being zero and the element is treated as being plane strain. Bearing in mind that LS-DYNA populates all 6 terms of the strain tensor (local **`Ezz`** being finite due to the need to conserve volume) this is a gross simplification, and these terms should be used with care.

D3PLOT will calculate the ratio of E2MIN_2D_PRINC_STRAIN / E2MAX_2D_PRINC_STRAIN which can be used to give an indication of the stress state:

E2D_PRINC_STRAIN_RATIO

A problem arises when trying to distinguish between biaxial tension and biaxial compression as in both cases they return a +ve value. In fracture predictions it is important to distinguish between the two so, whilst it is possbile that a compressive state could give a shear fracture, the most likely outcome would be buckling. Therefore, when both E2MAX_2D_PRINC_STRAIN and E2MIN_2D_PRINC_STRAIN are negative (or 0) a value of 1.1 is returned to indicate a compressive state.

The ratio is capped to -100 to avoid large values if E2MAX_2D_PRINC_STRAIN is relatively small compared to E2MIN_2D_PRINC_STRAIN

Components derived geometrically by D3PLOT

Thick shell volume and relative are calculated from the nodal coordinates:

VOLUME ***RV_RELATIVE_VOLUME***

Geometric components

These components are extracted from the topology, and can be output in **WRITE** :

MN_MATERIAL_NUMBER **LN_LIST_OF_NODES** **FE_FACING_ELEMENTS**

16.10.3. Averaged Nodal Data Components for Thick Shells

Averaged nodal data components for thick shells

All the nodal results from the `.PTF` file listed in Section 13.6.1 are available for contouring on thick shell elements (in 2D/3D plotting mode). They can also be extracted as averaged element data for use in **WRITE** and **XY_PLOT** .

Nodal contact force components from the `.CTF` file **cannot** be processed on thick shell elements.

16.10.4. Out of Plane (Z') Stress Tensor Components

Out of plane (Z') stress tensor components

Depending on their element formulation (elform on *SECTION_TSHELL) thick shells in LS-DYNA may or may not calculate the element through-thickness ($\sigma_{ZZ'}$) direct stress.

- Elform 1 (single point) and 2 (reduced 2x2 point) integration schemes are effectively extruded thin shells and do not calculate $\sigma_{ZZ'}$ which will always be zero.
- Elforms 3 onwards are closer to bricks than shells and develop a full 3d stress state, so $\sigma_{ZZ'}$ is computed.

The through thickness shear terms ($\tau_{YZ'}$ and $\tau_{XZ'}$) are calculated in all cases (just like thin shells).

Thus the local stress tensor will be one of

$$\begin{bmatrix} \sigma'_{xx} & & \\ \tau'_{yx} & \sigma'_{yy} & \\ \tau'_{zx} & \tau'_{zy} & 0 \end{bmatrix} \qquad \begin{bmatrix} \sigma'_{xx} & & \\ \tau'_{yx} & \sigma'_{yy} & \\ \tau'_{zx} & \tau'_{zy} & \sigma'_{zz} \end{bmatrix}$$

For shell-like formulations For brick-like formulations.

This means that principal stresses will not necessarily lie in the plane of the shell.

16.10.5. Out of Plane (Z') Strain Tensor Components

Out of plane (Z') strain tensor components

Shell-like thick shells do not **integrate** the through thickness strain ($\epsilon_{ZZ'}$), but this value may nevertheless be reported as non-zero. This is because the shell thickness may change, leading to a strain: its stiffness in this direction is based on volume preservation, but there is no stress associated with this.

Brick-like thick shells do calculate the through thickness strain.

Therefore principal strains also may not be confined to the plane of the element.

16.10.6. Frame of Reference: Computing the Local Coordinate System

Frame of reference: computing the local coordinate system

The local coordinate system used for thick shells is computed in much the same way as for shells, except that the average of the bottom (N1N2N3N4) and top (N5N6N7N8) faces is used to produce a "middle" face. The formulae are, as with shells:

X'	=	**N1N2**	approximately: the true **X'** vector is recomputed below
Z'	=	**N1N3 x N2N4**	(Where **x** is a vector cross-product)
Y'	=	**Z' x X'**	
X'	=	**Y' x Z'**	This final transformation is required to correct for any warping

In the case of a 6 noded thick tria Z' is obtained from N1N2 x N1N3

16.10.7. User-Defined Thick Shell Data Components

User-defined Thick shell data components

D3PLOT 9.3 permits you to create an unlimited number of user-defined data components for solids. These may be either scalar or tensor:

- Scalar components are just treated as numerical values with no known properties or orientation.

- Tensor components are assumed to be in the global system when created, and are subject to Frame of Reference transformation as described in Frame of reference: computing the local coordinate system above in exactly the same way as stresses and strains read from the database. The same derived components (principal, max shear, von Mises, etc) are also available.

16.11. Beam Element Results

Beam element results

Beams in LS-DYNA are always linear, 2-noded elements. There are several different formulations but the two principal types are:

Resultant:	Used for standard sections, and special "plastic" resultant formulations.
Integrated:	Also used for standard sections, but also for user-defined arbitrary sections.

Note that due to a bug in LS-DYNA the output sign convention used for the two types above is inconsistent up to and including LS-DYNA release 970, this problem has been fixed from LS-DYNA 971 onwards. This issue is described in beamip where controlling beam output is discussed. This has implications for beam plotting and when extracting cut section forces and moments through beam structures (see FORCES Computing forces and moments on the cutting plane).

16.11.1. "Basic" Components for All Beams

Basic components for all beams

All beams types generate a "basic" force and moment vector (even if some of them populate it entirely with zeros!). This is [`Fx, Fy, Fz, Mxx, Myy, Mzz`], where all results are written in the local axis system for beams. The D3PLOT components have the names:

FX_AXIAL_FORCE	**MXX_TORSIONAL_MOMENT**
FY_Y_SHEAR_FORCE	**MYY_BENDING_MOMENT**
FZ_Z_SHEAR FORCE	**MZZ_BENDING MOMENT**

The local axis system for beams is derived as follows:

If N1N2 is the vector from node 1 to node 2:

X'	=	**N1N2**	
Z'	=	**N1N2 x N1N3**	(Where x is the vector cross-product)
Y'	=	**Z' x X'**	

The "third" node (N3) is the orientation node for the beam, serving only to define its local Y' axis. It has no structural significance. If the representation of beam local axes during post-processing is important throughout an analysis you should consider defining separate "third" nodes for each beam element, and setting the < **nrefup** > field on the ***CONTROL_OUTPUT** card to update these nodes' coordinates.

16.11.2. "Extra" Components for Hughes-Liu Beams

Extra components for Hughes-Liu beams

If the **BEAMIP** flag on the ***DATABASE_EXTENT_BINARY** card is used "extra" data for **<beamip>** integration points in Hughes-Liu beams will be output. The 5 components for each point are:

XX_AXIAL_STRESS	**SP_PLASTIC_STRAIN**
YY_SHEAR_STRESS	**SAX_AXIAL_STRAIN**
ZZ_SHEAR_STRESS	

These values are calculated by LS-DYNA, and output in the relevant beam local axes.

The number of integration points worth of output is set by BEAMIP, so you will get (BEAMIP * 5) extra values for each beam in the output file. If a beam has fewer than BEAMIP integration points the "surplus" data slots will contain zeros, if a beam has more than BEAMIP integration points then output for points higher than BEAMIP will be lost.

16.11.3. "Extra" Components for Belytschko-Schwer Beams

Extra components for Belytschko-Schwer beams

If **BEAMIP** on the ***DATABASE_EXTENT_BINARY** card is set to 3 or more the following extra data components will be available for all Belytschko-Schwer beams: (the "distribution" components in italics, which show data variation down the beam, are computed internally by D3PLOT)

MY1_Y_BENDING_MOM_END_1 **MZ1_Z_BENDING_MOM_END_1**

MY2_Y_BENDING_MOM_END_2 **MZ2_Z_BENDING_MOM_END_2**

MYD_Y_MOM_DISTRIBUTION ***MZD_Z_MOM_DISTRIBUTION***

MMD_MOM_MAG_DISTRIBUTION

And if the Belytschko-Schwer beams use a resultant material formulation: (the "distribution" components in italics, which show data variation down the beam, are computed internally by D3PLOT)

RXX_PLASTIC_TORS_ROTN **RZ1_Z_PLASTIC_ROT_END_1**

RY1_Y_PLASTIC_ROT_END_1 **RZ2_Z_PLASTIC_ROT_END_2**

RY2_Y_PLASTIC_ROT_END_2 ***RZD_Z_MOM_DISTRIBUTION***

RYD_Y_ROT_DISTRIBUTION ***RMD_ROT_MAG_DISTRIBUTION***

PE1_PLASTIC_ENERGY_END_1 ***PED_PLASTIC_ENERGY_DIST***

PE2_PLASTIC_ENERGY_END_2 **EAX_AXIAL_ENERGY**

SAX_TOTAL_AXIAL_STRAIN

IE_INTERNAL_ENERGY

D3PLOT will also derive the following data components:

BED_BENDING_ENERGY_DENS = **(PE1 + PE2) / Length**

AED_AXIAL_ENERGY_DENS = **EAX / Length**

IED_INTERNAL_ENERGY_DEN = **IE / Length**

16.11.4. Notes on Beam Data

Notes on beam data

(1) Hughes-Liu (integrated) beams locate their integration point(s) at mid-span, and have a constant shear force and moment along their length.

The location and number of integration points through the thickness at mid span depends on the beam shape. See the *SECTION_BEAM keyword in the LS-DYNA manual for more information.

(2) Belytschko-Schwer (resultant) beams calculate the moment variation along the beam, so may have different Myy and Mzz terms at ends one and two. This presents a problem when only the basic force and moment vector is written since only one Myy and one Mzz term are output. **These are in fact the values at end 1.** So if you have a cantilever fixed at end 2, with a point load at end 1, you will not see any moment in it if you only plot the basic Myy or Mzz data components (although the moment will be there and it will behave correctly). Furthermore, the sign of the end 1 Myy and Mzz moments written to the extra data slots is opposite to the sign of the basic moment vector Myy and Mzz moments so care must be taken in interpreting the direction of moments when switching between basic and extra data component moments.

The best solution to this problem is to write the "extra" data since, as is shown in the table below, separate end 1 and end 2 moments are then written to file.

(3) At present there is no way to tell from the database whether the "extra" data is for Belyschko-Schwer (resultant) or Hughes-Liu (integrated) beams. So both sets of options may be extracted from the same dataset.

The table below shows how the different sets of data overlap in these data slots: ***it is your responsibility to interpret your data correctly.***

How resultant and integrated "extra" beam data components overlap in the LS-DYNA output file

LS-DYNA re-uses the same "extra" data slots, as stipulated by BEAMIP on *DATABASE_EXTENT_BINARY, to contain integrated and resultant beam data. There will be 5 * BEAMIP such slots, and they will contain the following data components:

Slot	Resultant	Integrated	
1	Total axial strain	Int pt 1	XX axial stress
2	Plastic energy end 1		YZ shear stress

3	Plastic energy end 2		ZX shear stress
4	Y plastic rotation end 1		Plastic strain
5	Y plastic rotation end 2		Axial strain
6	Z plastic rotation end 1	Int pt 2	XX axial stress
7	Z plastic rotation end 2		YZ shear stress
8	Y bending moment end 1		ZX shear stress
9	Y bending moment end 2		Plastic strain
10	Z bending moment end 1		Axial strain
11	Z bending moment end 2	Int pt 3	XX axial stress
12	Axial energy		YZ shear stress
13	Internal energy		ZX shear stress
14	Plastic torsional rotation		Plastic strain
15	Spare		Axial strain
16	Slots #16 onwards not used	etc for further int pts in blocks of 5 values	

If a model contains a mixture of integrated and resultant beams, and BEAMIP is >= 3 (that is 15 or more "extra" data values are written per beam), before v21 D3PLOT cannot know how to interpret the data, that is which column to use in the table above for any given beam. As a result it makes both sets of data component available for plotting, and it is up to you do choose the correct ones for the beams in your model. From v21 D3PLOT will use data in the ZTF file, if available, to identify beam element type (integrated or resultant) and will, by default, only plot integrated beam components on integrated beam elements and only plot resultant beam components on resultant beam elements. If the ZTF file is not available the behaviour will be as for earlier versions of D3PLOT. It is possible to switch the behaviour to the old behaviour, i.e. to ignore beam type when contouring, by using the Ignore Beam Type switch in Beam Symbols. Alternatively you can set the preference d3plot*ignore_beam_type:ON.

You need to exercise extreme care when interpreting results in this situation and it suggested either that you do not mix the two beam types in a model, or if this is necessary then you assign the PART numbers of each type of beam to different label ranges, or perhaps use some common identifier in the PART names, making it easy to control what is displayed by blanking.

16.12. Contact Segment Results

Contact segment results

This section describes the results available for individual contact segments, as distinct from those for surfaces as a whole.

Contact segment results are written to the `.CTF` file, so if this file is missing it will not be possible to visualise or process contact surface results.

16.12.1. What are Contact Segments?

What are contact segments?

They are not really elements, although it is convenient to treat them as such within D3PLOT. They are 3 or 4 noded areas over which contact is calculated, and which must lie on "real" structural elements underneath.

Contact forces are calculated at nodes, and then averaged over the area of their connected segments to give contact "stress": really these value are "pressure" not "stress" (although the units are the same).

Users should be aware that contact forces in LS-DYNA are calculated from the repulsion forces required to stop nodes penetrating surfaces, and that this is an inherently noisy process since penetrations - and hence forces - tend to oscillate. Therefore contact forces on individual segments at a given state should be treated as a snapshot of a dynamic process, and not necessarily a good indication of the mean contact force averaged over a longer time period.

16.12.2. Components Written by LS-DYNA for Contact Segments

Components written by LS-DYNA for contact segments

Contact segment "stress" values written by LS-DYNA at segments:

CN_CONTACT_NORMAL	Stress normal to surface;
CT_CONTACT_TANGENTIAL	Resultant stress in plane of surface;
CX_CONTACT_X	In-plane local X stress;
CY_CONTACT_Y	In-plane local Y stress;

Contact forces written by LS-DYNA at nodes, averaged by D3PLOT over segments:

XG_GLOBAL_X_FORCE	***XL_LOCAL_X_FORCE***	***FM_FORCE_MAGNITUDE***
YG_GLOBAL_Y_FORCE	***YL_LOCAL_Y_FORCE***	
ZG_GLOBAL_Z_FORCE	***ZL_LOCAL_Z_FORCE***	

Local forces are transformed to reflect the orientation of a segment. But since the nodal forces on which they are based also have contributions from adjacent segments they should be regarded as approximate.

Versions of LS-DYNA may also write:

CG_CONTACT_GAP	Contact gap at nodes N1 to N4 (possibly from LS970 onwards)
CE_ENERGY_DENSITY	}
	} Possibly from LS971 onwards
CPP_PEAK_PRESSURE	}
CPT_TIME_TO_PEAK_PRE	}

16.12.3. Geometric Components Calculated by D3PLOT

Geometric components calculated by D3PLOT

ON_OUTWARD_NORMAL	Special geometric component to show segment orientation.
CA_CONTACT_AREA	The calculated area of each contact segment

16.12.4. Results in CTF File for Other Analysis Types

Results in CTF file for other analysis types

More recent versions of ls-dyna, typically LS971R5 onwards, can also use the Contact Force File (also known as Interface Force File) to contain analysis-specific data at segments. This feature is not well documented, so documentation is sketchy, however the following analysis types may produce the following components:

Analysis type	Component names	Meaning

CPM (airbag particle method) **DEM** (unknown) **ALE**	**IPR_PRESSURE** **IFX_X_FORCE** **IFY_Y_FORCE** **IFZ_Z_FORCE** **IFM_FORCE_MAG**	Contact pressure Contact forces in the global system
ALE only	**ISS_SLIP_SPEED** **ISX_SLIP_X_VEL** **ISY_SLIP_Y_VEL** **ISZ_SLIP_Z_VEL** **ISM_SLIP_VEL_MAG**	Slipping speed data

Because this file seems to be growing in both content and usage, and the documentation can lag behind this, D3PLOT also provides access to all of its data components as "raw" scalar data of unknown type. The data component names are generic:

Component names	Meaning
IF1_INTERFACE_1 **IF2_INTERFACE_2**	1st data component 2nd data component
IFn_INTERFACE_n	nth data component

The value of **n** above is determined by the file contents, but values of 4, 8, 16, 17, 21, and 23 are typical.

If you use these "raw" data components the interpretation of the data is your responsibility! Oasys Ltd ***may*** be able to advise about their content - please ask.

16.12.5. Results for Whole Surfaces

Results for whole surfaces

D3PLOT can also sum up results for contact surfaces, and the results can be viewed numerically in **WRITE** and **XY_DATA** . These values are the numerical sum of the relevant component for all segments in the surface.

When LS-DYNA writes contact surface results to the .CTF file it distinguishes between Surface A and B sides of each contact, therefore D3PLOT allows you to report results for contact surfaces as follows:

<table>
<tr><td>A side</td><td>Results are computed from segments on the surface A side</td></tr>
<tr><td>B side</td><td>Results are computed from segments on the surface B side</td></tr>
<tr><td>Surface as a whole where the summary results from A and B sides are within 5% by magnitude.

(A and B sides should be equal in magnitude, and directional components opposite in sign.)</td><td><table><tr><td>Directional (eg X force)</td><td>(B side - A side) * 0.5</td></tr><tr><td>Magnitude values</td><td>(A side + B side) * 0.5</td></tr></table>
Therefore the sign of the output of directional values is that of the surface B side.</td></tr>
<tr><td>Surface as a whole where summary results from A and B surface sides differ by more than 5% by magnitude. (Typically the single-surface case, or "nodes to" types with no A surface segments.)</td><td>Whichever side has the greater value by magnitude, with no sign change applied.</td></tr>
</table>

16.12.6. How LS-DYNA Calculates and D3PLOT Processes Contact "Stresses"

How LS-DYNA calculates and D3PLOT processes contact stresses

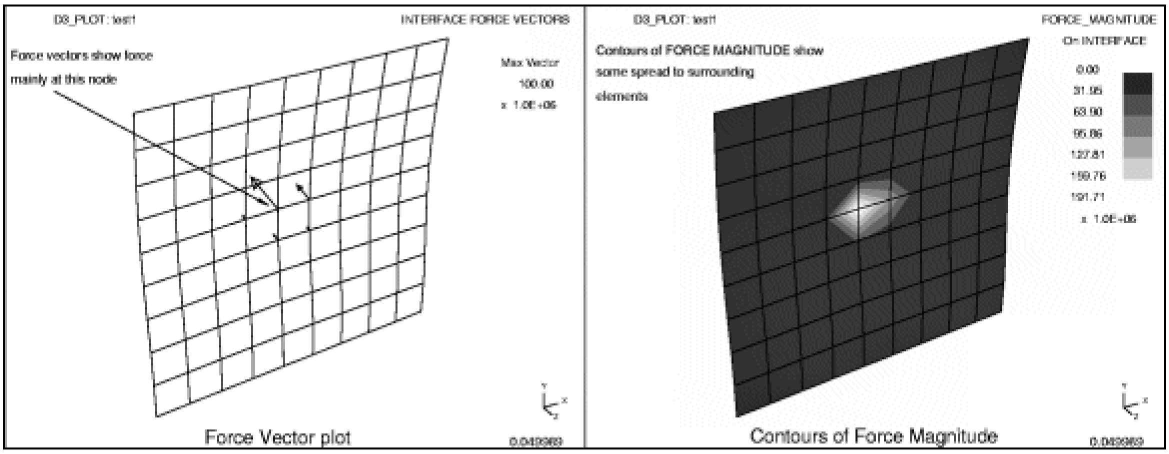
IMPORTANT:

The "stresses" on interfaces will always be lower than the true values in the elements, and they will also be spread over a wider area.

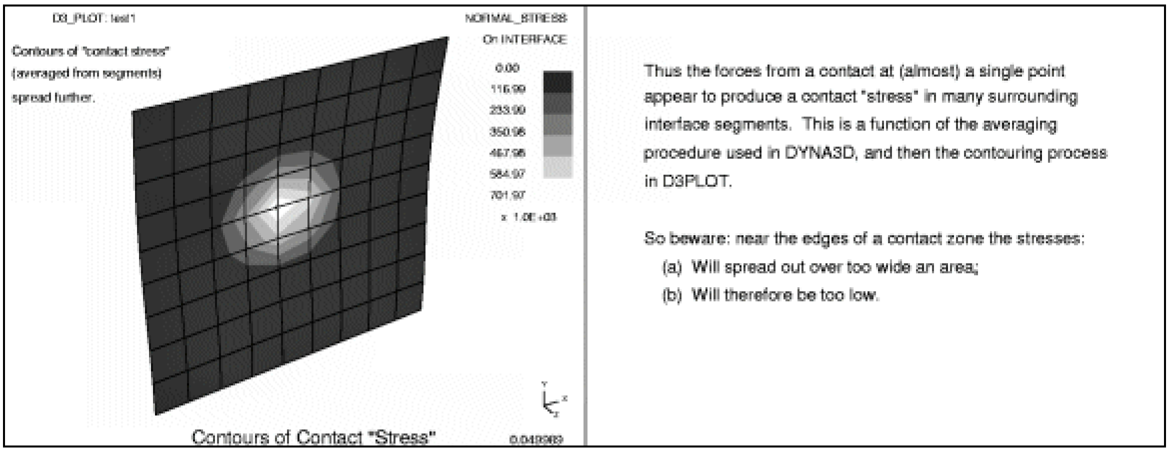
This is because of the way the penalty force and contouring algorithms work. Consider the following example (see the three figures below)

a) A single node (say on the surface B side) touches part of the A surface. (A and B sides are inter-changeable in this example.)

b) The reaction forces on the A side are distributed among the four nodes on the A side segment.



c) Contact stress is then computed over the sphere of influence of each node on the A surface, and stress is assigned to each contact segment.



d) Contouring which, in D3PLOT averages over elements, spreads the stress over a further region of contact segments.

Thus the contact at a single point has generated an apparent stress spread over twenty five elements - clearly this is not correct. To get round this you can:

- Plot "force" components (eg **FM_FORCE_MAGNITUDE**) which omit step (c) above, so force is spread only over nine segments in this example.
- Plot force vectors (**VECTOR** command) which will only show force arrows at the four nodes on the A side segment.

However in real examples the problem is less severe: the genuine contact is usually spread over several segments and it is only around the borders of the contact region that stresses spread out too far.

Nevertheless contact "stresses" should not be treated as more than approximate contact pressures, and in particular they must not be expected to be the same as (true) stress in the underlying elements.

16.13. Smooth Particle Hydrodynamic (SPH) Data Components

Smooth Particle Hydrodynamic (SPH) Data components

From release 9.3 D3PLOT will process SPH elements if present in the database.

16.13.1. SPH Membership of PARTs

SPH membership of PARTs

SPH elements belong to PARTs in exactly the same way as solids, shells and beams; so D3PLOT processes them "by part", and makes their part-based data components available in exactly the same way.

In addition it will be clear from the tables below that the stress and strain SPH database components are the same as solid and shell ones, so they are contoured alongside them in 2D3D plotting mode, fitting in neatly with PART membership.

16.13.2. SPH Data Organisation

SPH Data organisation

SPH elements can be thought of as having a single integration point at their centre, and all directional tensor components are written out in the global model system. The global/local frame of reference transformation has no effect on these results, and the current surface/integration point setting also does not affect them.

SPH elements can be deleted, as with other elements. D3PLOT detects this and removes deleted elements from the plot.

16.13.3. The Results Available for SPH Elements

The results available for SPH elements

SPH elements write out the following block of 18 values per element to the `.PTF` file:

- Radius of influence (1 value, used to determine its size when drawn)
- Pressure (1 value)

- Stress tensor (6 values)
- Plastic strain (1 value)
- Density (1 value)
- Internal energy (1 value)
- Number of neighbours (1 value)
- Strain tensor (6 values)

Experience suggests that the above data components are written regardless of any flags on `*DATABASE_EXTENT_BINARY` or other control cards.

The following tables show the raw data component names, and also those derived by D3PLOT.

Symmetric stress tensor

X_DIRECT_STRESS	**XY_SHEAR_STRESS**
Y_DIRECT_STRESS	**YZ_SHEAR_STRESS**
Z_DIRECT_STRESS	**ZX_SHEAR_STRESS**

$$\begin{bmatrix} \sigma_{xx} & & \\ \tau_{yx} & \sigma_{yy} & \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$

This is written in the global cartesian coordinate system.

The stress components that can be derived this by D3PLOT are:

MAX_PRINC_STRESS	***MAX_DEV_PRINC_STRESS***	***VON_MISES_STRESS***	***PRESSURE***
MID_PRINC_STRESS	***MID_DEV_PRINC_STRESS***	**SIGNED_VON_MISES_STRESS**	***TRIAXIALITY***
MIN_PRINC_STRESS	***MIN_DEV_PRINC_STRESS***	***MAX_SHEAR_STRESS***	***LODE_ANGLE***
YIELD_UTILISATION_FACTOR	***YIELD_UTILISATION_PERCENTAGE***	***LODE_PARAMETER***	***LODE_PARAMETER_ALT***

Effective plastic strain $[\epsilon_p]$

PLASTIC_STRAIN

The effective plastic strain is always output, as above. It has no intrinsic direction.

Directional strain tensor

Unlike other element types this strain tensor is written unconditionally, regardless of the value of **STRFLG** on the ***DATABASE_EXTENT_BINARY** card.

$$\begin{bmatrix} \epsilon_{xx} & & \\ \epsilon_{yx} & \epsilon_{yy} & \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix}$$

SX_DIRECT_STRAIN **SXY_SHEAR_STRAIN**

SY_DIRECT_STRAIN **SYZ_SHEAR_STRAIN**

SZ_DIRECT_STRAIN **SZX_SHEAR_STRAIN**

And the strains derived from these by D3PLOT

SMAX_PRINC_STRAIN ***SVON_MISES_STRAIN*** ***SAV_AVERAGE_STRAIN***

SMID_PRINC_STRAIN ***SMAX_SHEAR_STRAIN***

SMIN_PRINC_STRAIN ***PEMAG_PLAST_STRN_MAG***

Further SPH-only components

RADIUS **NUM_NEIGHBOURS**

VOLUME **ENERGY**

DENSITY **PRESSURE**

It is not clear why PRESSURE is written separately, rather than being deduced from the stress tensor. D3PLOT uses this value rather than (Sx + Sy + Sz) / -3.0

The following are derived from the above:

VOLUME (from radius)

16.14. Airbag Particle (ABP) Data Components

Airbag Particle (ABP) data components

From release 9.3 D3PLOT processes Airbag Particle elements.

These can be thought of as small spherical particles which are emitted from an inflator to apply pressure to an airbag, giving a more realistic pressure distribution than a control volume. Each inflator contains one or more gas generators, which emit particles as the analysis progresses. This means that - in effect - particles are "born" when they first appear in the analysis, and while they don't "die" as such they may pass through the fabric of the bag (or through a vent hole) and cease to take an active part in the analysis. Therefore the number of particles is initially zero, and their quantity increases as they pop into existence as the analysis progresses.

The way particles act upon the fabric of the bag is effectively to make contact with it and hence apply force, mimicking the true behavior of actual gas particles - albeit on a much larger scale.

Airbag particles are treated as elements within D3PLOT, and such data values that are as contourable are displayed in the 2D3D plotting mode. However as will be clear from the tables below these particles act more like small rigid balls and don't have element data as such, so the main purpose in plotting them is to see how they are inflating the airbag and "contouring" their data is not usually very helpful.

At the time of writing (September 2008) the information available about this feature is limited, and the information below has been largely reverse-engineered from what is present in databases written from example analyses run in LS-DYNA 971R4. It is likely that this feature will be developed further, and generate more information in the future.

16.14.1. ABP Membership of "Airbags"

ABP membership of "Airbags"

Within D3PLOT airbag particles belong to "Airbags" in exactly the same sense that solids, shells and beams belong to "Parts". (The "airbag" is the Control Volume definition in the input deck.) It is not possible to tell from the .ptf file what the fabric elements making up a bag are, so it cannot be drawn explicitly, however being able to select and process "by airbag" fits neatly into the logic of the programme, and makes it easy to select and control these elements.

"Per airbag" data can be plotted (all particles in a bag getting the same contour value), and can also be displayed numerically in **WRITE** and **XY_DATA** . At present the data components available at "Airbag" level are:

ANP_ABAG_NUM_PARTICLES	The number of "live" particles in this bag. A particle airbag has a maximum number of potential particles defined (field `NP` on the `*AIRBAG_PARTICLE` card), and as the inflator fires and gas is expelled so the number of "live" particles active in the bag rises from an initial value of

	zero to some number (typically) less than this maximum value.
AVOL_AIRBAG_VOLUME	The current bag volume as reported by LS-DYNA

D3PLOT also calculates the following components for the airbag as a whole by summation of individual particle data. These components are a single value calculated for the whole bag, when contoured this single value is used for all particles.

ABE_AIRBAG_TOTAL_ENERGY	This is the translational kinetic energy of all the gas particles in the bag, computed from the sum of: (**Translational Kinetic Energy**) + **(Spin Energy**) for all "live" particles.
ABK_AIRBAG_TRANS_KE	This is the sum of translational kinetic energy for all the "live" particles in the bag.
ABK_AIRBAG_SPIN_ENERGY	This is the sum of spin energy for all the "live" particles in the bag.
ABP_AIRBAG_PRESSURE	This is (**2/3 *** ***ABE_AIRBAG_KINETIC ENERGY*** **/** **AVOL_AIRBAG_VOLUME**) This is a single value calculated for the whole bag, when contoured this single value is used for all particles. ***NOTE*** : This is not the same as the "mapped" **PARTICLE_PRESSURE** data component described below. Note that the pressure will not include any atmospheric component (field `PATM` on the `*AIRBAG_PARTICLE` card) until the inflator fires since no information is available until some "live" particles are present.

A description of the `*AIRBAG_PARTICLE` method, giving the theory and formulae used to derive the components above, may be found here http://www.impetus.no/inc/openitem.asp?id=36776&nid=696

16.14.2. ABP Data Components

ABP data components

The following data components are written from LS-DYNA 971R4 and may be "contoured" in 2D3D plotting mode as data components. "Contouring" is a misnomer in this context, since each particle is independent and has a single value, however the display of this value can still be useful.

Current coordinates

CX_CURRENT_X_COORD **CY_CURRENT_Y_COORD** **CZ_CURRENT_Z_COORD**

`Displacements (Derived from <current> - <undeformed>)`

DX_X_DISPLACEMENT ***DY_Y_DISPLACEMENT*** ***DZ_Z_DISPLACEMENT***

DR_DISP_RESULTANT

Velocities

VX_X_VELOCITY **VY_Y_VELOCITY** **VZ_Z_VELOCITY**

VR_VEL_RESULTANT

Other values written directly by LS-DYNA

MASS

RADIUS

NS_DIST Distance to nearest bag segment, set to 1e10 if "far" away from a segment.

GAS_ID The gas number

LEAKAGE **0** = inside bag
1 = escaped due to porosity
2 = escaped through vent
3 = MPP error

SPIN_ENERGY

TRANS_ENERGY

VOLUME (from **RADIUS**)

DENSITY (from **MASS** / ***VOLUME***)

Other values calculated by D3PLOT

Data mapping (see Section 4.4.2.8) can be used to calculate further data components for airbag particles:

PARTICLE_DENSITY	Simply the number of particles per unit volume, calculated by mapping particles and dividing the number of particles in each cell by the volume of that cell. So it is a measure of the density of particles in space and, like particle pressure below, suffers from the problem that it will be an under-estimate near the edge of the bag where fabric material cuts mapping cell. ***This is not the same as the DENSITY component above*** , which shows the mass of each particle divided by that particle's volume.
PARTICLE_PRESSURE	Crudely the pressure "near" a small number of airbag partices is equal to 2/3 the sum of the translational energies of those particles divided by the "near" volume. The detailed theory is beyond the scope of this manual, but is available from Oasys Ltd on request. Data mapping, in which the volume of space around the particles is divided in to cells, and values are calculated for each cell, can be used to derive an approximate pressure in each cell. This in turn can be assigned to all particles in the cell and contoured as pressure, and an ISO plot can also be performed to show the approximate pressure distribution. This process is not accurate near the edges of the bag where the fabric will intersect a cell, since the volume of that cell is over-estimated and hence the pressure within it is under-estimated. Nevertheless it can give a reasonable display of gas pressure and flow.
PARTICLE_VELOCITY	A similar process, this time calculating vector rather than scalar data, can be used to average particle velocities in mapping cells, giving an approximate average velocity for each cell. A vector plot of this can be used to plot the approximate gas flow direction and velocity.

16.14.3. Nodes on ABP Elements: VISFLG on *AIRBAG_PARTICLE Card

Nodes on ABP elements: VISFLG on *AIRBAG_PARTICLE card

The documentation on the **`*AIRBAG_PARTICLE`** card suggests that **`VISFLG`** must be turned on in order to see airbag particles. This is misleading: airbag particle data is written to the .ptf file regardless if present in the analysis, and this flag simply turns on the output of nodes coincident with these elements.

Since D3PLOT 9.3 onwards will visualise ABP elements without needing these nodes it is recommended that you turn **`VISFLG`** off in order to reduce the output database size.

16.15. Discrete Spherical Element (DES) Data Components

Discrete Spherical element (DES) data components

From release 12.0 D3PLOT processes Discrete Spherical elements

16.15.1. DES Membership of PARTs

DES membership of PARTs

DES elements belong to PARTs in exactly the same way as solids, shells and beams; so D3PLOT processes them "by part", and makes their part-based data components available in exactly the same way.

16.15.2. SPH Data Organisation

SPH Data organisation

DES elements can be thought of as having a single integration point at their centre, and all directional tensor components are written out in the global model system. The global/local frame of reference transformation has no effect on these results, and the current surface/integration point setting also does not affect them.

DES elements can be deleted, as with other elements. D3PLOT detects this and removes deleted elements from the plot.

16.15.3. The Results Available for DES Elements

The results available for DES elements

DES elements write out a variable length block of data that can contain all or a subset of the following 13 values per element to the **`.PTF`** file:

- Radius (1 value)

- Mass (1 value)
- Inertia (1 value)
- Active flag (1 value)
- Stress tensor (6 values)
- Volumetric strain (1 value)
- Damage Parameter (1 value)
- Internal energy (1 value)

The following tables show the raw data component names, and also those derived by D3PLOT.

Symmetric stress tensor:

X_DIRECT_STRESS	XY_SHEAR_STRESS
Y_DIRECT_STRESS	YZ_SHEAR_STRESS
Z_DIRECT_STRESS	ZX_SHEAR_STRESS

$$\begin{bmatrix} \sigma_{xx} & & \\ \tau_{yx} & \sigma_{yy} & \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$

This is written in the global cartesian coordinate system.

The stress components that can be derived this by D3PLOT are:

MAX_PRINC_STRESS	*MAX_DEV_PRINC_STRESS*	*VON_MISES_STRESS*	*PRESSURE*
MID_PRINC_STRESS	*MID_DEV_PRINC_STRESS*	*SIGNED_VON_MISES_STRESS*	*TRIAXIALITY*
MIN_PRINC_STRESS	*MIN_DEV_PRINC_STRESS*	*MAX_SHEAR_STRESS*	*LODE_ANGLE*
YIELD_UTILISATION_FACTOR	*YIELD_UTILISATION_PERCENTAGE*	*LODE_PARAMETER*	*LODE_PARAMETER_ALT*

Further DES-only components

RADIUS	VOLUMETRIC_STRAIN
MASS	DAMAGE_PARAMETER
INERTIA	INTERNAL_ENERGY
ACTIVE	

The following are derived from the above:

VOLUME (from radius)

16.16. SPRINGER/DAMPER Components

SPRING/DAMPER components

No spring/damper data components are written to the D3PLOT (PTF) file.

D3PLOT uses data written to the ZTF file (generated by Oasys Ltd. PRIMER) to draw spring/damper elements and for geometric data components. D3PLOT can also extract additional spring/damper data components from the LSDA (binout) file if present.

16.16.1. Spring/Damper LSDA (binout) Data Components

Spring / Damper LSDA (binout) Data Components

As the output frequency of data to the LSDA binout file can be different to the output frequency of the D3PLOT (PTF) file there is no guarantee that data in LSDA file will be available at exactly the same time as the states in the PTF file. When plotting data from the LSDA file D3PLOT will find the output state closest to the PTF file time. If the nearest LSDA time is not within 10% of the PTF output frequency then no data will be available for Springs/Dampers at that PTF time.

The following data components are only written to the LSDA (binout) file if the *DATABASE_DEFORC option is set.

SPRING_FORCE	Translational Springs/Dampers
SPRING_ELONGATION	Translational Springs/Dampers
SPRING_MOMENT	Rotational Springs/Dampers
SPRING_ROTATION	Rotational Springs/Dampers

16.16.2. Spring/Damper Geometric Data Components

Spring/Damper Geometric Data Components

D3PLOT cannot currently extract time-dependent data for these element types, (except normal force on stonewalls), so only geometric data components are available for them. These data are extracted from the `.xtf` file if present, or from the `.ztf` file if not. If neither file is present then these items will not be processed.

LN_LIST_OF_NODES TYPE_OF_SPRING MN_MATERIAL_NUMBER

SUMMARY

In addition if the **BEAM** flag on the ***DATABASE_BINARY_D3PLOT** card may be used to write spring results into beam "slots" in the .ptf file as follows:

BEAM = 0 (default)	Extra beam elements are added to the .ptf file using the spring/damper topology, and Global [Fx, Fy, Fz, Fr] forces are written in the Fx, Fy, Fz, Myy data "slots" for beams
BEAM = 1	No extra beams are written
BEAM = 2	Extra beam elements are written as for the "0" case above, but only the resultant force is written to the Fx "slot".

If these extra beams are present in the database D3PLOT has no way of knowing whether they are genuine beams, or springs masquerading as beams. Therefore if you use this option treat your results with care.

16.17. SEATBELT Components

SEATBELT components

No seatbelt data components are written to the D3PLOT (PTF) file.

D3PLOT uses data written to the ZTF file (generated by Oasys Ltd. PRIMER) to draw seatbelt elements and for geometric data components. D3PLOT can also extract additional seatbelt data components from the LSDA (binout) file if present.

16.17.1. SEAT_BELT, RETRACTOR and SLIP_RING LSDA (binout) Data Components

SEAT_BELT, RETRACTOR and SLIP_RING LSDA (binout) Data Components

As the output frequency of data to the LSDA binout file can be different to the output frequency of the D3PLOT (PTF) file there is no guarantee that data in LSDA file will be available at exactly the same time as the states in the PTF file. When plotting data from the LSDA file D3PLOT will find the output state closest to the PTF file time. If the nearest LSDA time is not within 10% of the PTF output frequency then no data will be available for Seatbelts at that PTF time.

The following data components are only written to the LSDA (binout) file if the *DATABASE_SBTOUT option is set.

BELT_FORCE	Seatbelts
BELT_LENGTH	Sliprings
SLIP_RING_PULL_THROUGH	Retractors
RETRACTOR_FORCE	
RETRACTOR_PULL_OUT	

16.17.2. SEAT_BELT, RETRACTOR and SLIP_RING Geometric Data Components

SEAT_BELT, RETRACTOR and SLIP_RING Geometric Data Components

MN_MATERIAL_NUMBER **LN_LIST_OF_NODES** **SUMMARY**

Since seatbelt elements are really discrete elements inside LS-DYNA the BEAM flag on *DATABASE_BINARY_D3PLOT will also result in beams being written out for seatbelt elements as described in Spring/Damper Geometric Data Components above.

16.18. SPOTWELD Components

SPOTWELD components

No spotweld data components are written to the D3PLOT (PTF) file.

D3PLOT uses data written to the ZTF file (generated by Oasys Ltd. PRIMER) to draw spotwelds. D3PLOT can also extract spotweld data components from the LSDA (binout) file if present.

16.18.1. SPOTWELD LSDA (binout) Data Components

SPOTWELD LSDA (binout) Data Components

As the output frequency of data to the LSDA binout file can be different to the output frequency of the D3PLOT (PTF) file there is no guarantee that data in LSDA file will be available at exactly the same time as the states in the PTF file. When plotting data from the LSDA file D3PLOT will find the output state closest to the PTF file time. If the nearest LSDA time is not within 10% of the PTF output frequency then no data will be available for Spotwelds at that PTF time.

The following data components are only written to the LSDA (binout) file if the *DATABASE_SWFORC option is set.

SPOTWELD_FORCE	**SPOTWELD_FAILURE**	**SPOTWELD_TORSION**
SPOTWELD_SHEAR	**SPOTWELD_FTIME**	**SPOTWELD_LENGTH**

The following additional data components are only written to the LSDA (binout) file if the *DATABASE_DCFAIL option is set.

DCFAIL_FAILURE	**DCFAIL_SHEAR**	**DCFAIL_AREA**
DCFAIL_NORMAL	**DCFAIL_BENDING**	

16.19. X-SECTION Components

X-SECTION components

No spotweld data components are written to the D3PLOT (PTF) file.

D3PLOT uses data written to the ZTF file (generated by Oasys Ltd. PRIMER) to draw X-Sections defined using the *DATABASE_CROSS_SECTION option in LS-DYNA. D3PLOT can also extract X-Section data components from the LSDA (binout) file if present.

16.19.1. X-SECTION LSDA (binout) Data Components

X-Section LSDA (binout) Data Components

As the output frequency of data to the LSDA binout file can be different to the output frequency of the D3PLOT (PTF) file there is no guarantee that data in LSDA file will be available at exactly the same time as the states in the PTF file. When plotting data from the LSDA file D3PLOT will find the output state closest to the PTF file time. If the nearest LSDA time is not within 10% of the PTF output frequency then no data will be available for X-Sections at that PTF time.

The following data components are only written to the LSDA (binout) file if the *DATABASE_SECFORC option is set.

X_FORCE	X_MOMENT	RESULTANT_FORCE
Y_FORCE	Y_MOMENT	RESULTANT_MOMENT
Z_FORCE	Z_MOMENT	AREA

16.20. LOAD PATH Components

LOAD PATH components

No spotweld data components are written to the D3PLOT (PTF) file.

LOADPATHS are an designed as an easier way to see how the load is transmitted through a structure. They are created in Oasys Ltd. PRIMER and are defined by creating a path through multiple *DATABASE_CROSS_SECTION definitions. Each LOADPATH consists of multiple LOADPATH segments.

D3PLOT uses information in the ZTFILE to locate and draw LOADPATHS.

16.20.1. LOADPATH Data Components

LOADPATH Data Components

The data components for each LOADPATH segment are derived from the forces and moments output for the X-Sections at each end of of the LOADPATH segment by rotating the X-Section forces and moments into the vector defined between the X-Section centroids at each end of the segment.

As LOADPATH components are derived from X-Section forces and moments the *DATABASE_SECFORC option needs to be set for the following components to be available.

FX_AXIAL_FORCE	MYY_BENDING_MOMENT	RESULTANT SHEAR
FY_SHEAR_FORCE	MZZ_BENDING_MOMENT	RESULTANT_MOMENT
FZ_SHEAR_FORCE	MXX_TORSIONAL_MOMENT	

In addition to the derived data components the following geometric based components can also be plotted which used the X-section centroid coordinates.

BX_BASIC_X_COORD	CX_CURRENT_X_COORD
BY_BASIC_Y_COORD	CY_CURRENT_Y_COORD
BZ_BASIC_Z_COORD	CZ_CURRENT_Z_COORD
LL_LOADPATH_LENGTH	Loadpath Length is measured along the vectors joining the X-section centroids

The following "raw" X-Section data components are also available.

X_FORCE	X_MOMENT	RESULTANT_FORCE
Y_FORCE	Y_MOMENT	RESULTANT_MOMENT
Z_FORCE	Z_MOMENT	

16.21. Data Components for Loads

Data components for loads

LOADS :

D3PLOT extracts data for loads from the .ztf file. If this file is not present then these items will not be processed.

16.21.1. LOAD Components

LOAD components

LOAD_FORCE	LOAD_LINE_LOAD	LOAD_PRESSURE
LOAD_MOMENT		

16.22. Data Components for Other Entity Types

Data components for other entity types

LUMPED_MASSES , **SPRINGS** , **SEAT_BELTS** etc, **JOINTS** , **STONEWALLS** :

D3PLOT cannot currently extract time-dependent data for these element types, (except normal force on stonewalls), so only geometric data components are available for them. These data are extracted from the `.xtf` file if present, or from the `.ztf` file if not. If neither file is present then these items will not be processed.

16.22.1. LUMPED-MASS Components

LUMPED-MASS components

LN_LIST_OF_NODES   CE_CONNECTED_ELEMENTS   MASS

SUMMARY

16.22.2. JOINT Components

JOINT components

LN_LIST_OF_NODES   TYPE_OF_JOINT   STIFFNESS

SUMMARY

16.22.3. STONEWALL Components

STONEWALL components

NORMAL_FORCE   SUMMARY

Notes:

- The **SUMMARY** components above are the most useful since they list all relevant data for each element type.
- Spring, seat-belt and stonewall transient data may be extracted from the `.XTF` file via the T/HIS time-history plotting programme.
- Velocity of moving stonewalls can be deduced from the velocities of the optional extra nodes that can be placed on such stonewalls.

16.23. Data Components for Multiphysics Solvers

Data components for Multiphysics solvers

In order to display results from these solvers an additional file called "multiphysics.components" must be present in the directory containing the D3PLOT executable.

In version 12 results from all 3 solvers can be plotted using any of the standard plotting modes (CT, SI, LC, ISO, CL, VEC - See Drawing commands that plot data for more details) but support in other menus is limited. At present ICFD, CESE and EMAG results are not available in either the WRITE (see WRITE Listing Numerical Data to Screen and/or File) or XY-DATA (see XY_DATA Drawing Numerical Data as XY Plots and/or Writing it to File) menus.

16.23.1. ICFD Components

ICFD components

ICFD results are available for both surfaces and volumes, some components (eg. fluid velocity) exist for both surface and volumes while others are available for either surfaces or volumes. As the surface and volume results are output separately by LS-DYNA then the user must select either an ICFD volume component or an ICFD surface component, this means that it is not currently possible to plot fluid velocity on both surfaces and volumes at the same time even though fluid velocity is output for both.

ICFD results are output at the surface and volume nodes.

ICFD Surface Scalar Data Components	**FLUID PRESSURE** **SURFACE SHEAR**
ICFD Surface Vector Data Components	**FLUID VELOCITY**
ICFD Volume Scalar Data Components	**FLUID PRESSURE** **FLUID VORTICITY** **Q CRITERION** **AVERAGE PRESSURE**
ICFD Volume Vector Data Components	**FLUID VELOCITY**

16.23.2. CESE Components

CESE components

CESE results can be plotrted fopr both surfaces and volumes but at present the only CESE data components are for Volumes.

CESE results are output at the centre of elements.

CESE Volume Scalar Data Components	**DENSITY** **PRESSURE** **TEMPERATURE** **TOTAL ENERGY** **VOID FRACTION** **SCHLIEREN NUMBER**
CESE Volume Vector Data Components	**FLUID VELOCITY** **VORTICITY**

16.23.3. EMAG Components

EMAG components

EMAG results are available for both surfaces and volumes. EMAG volume data components are plotted on the underlying structural elements.

EMAG Surface Vector Data Components	**SURFACE CURRENT** **MAGNETIC FIELD** **LORENTZ FORCE**
EMAG Volume Scalar Data Components	**ELECTRICAL CONDUCTIVITY** **OHM HEATING POWER**

EMAG Volume Vector Data Components

SCALAR POTENTIAL

CURRENT DENSITY

ELECTRIC FIELD

MAGNETIC FIELD

LORENTZ FORCE

VECTOR POTENTIAL

16.24. Theory and Formulae

Theory and Formulae

This section describes some of the theory and equations use to process data components in D3PLOT.

16.24.1. Manipulations of Stress Tensor Components

Manipulations of stress tensor components

We have adopted the tensor notation for global stresses :

$$[S] = \begin{bmatrix} \sigma_{xx} & & \\ \tau_{yx} & \sigma_{yy} & \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$ (Symmetric: Upper triangle = lower triangle)

Where σ_{XX}, σ_{YY}, σ_{ZZ} are the "direct" stresses, and τ_{XY}, τ_{YZ}, τ_{XZ} the shear stresses.

By convention the ij suffices on these terms mean "stress in the direction i on a plane of constant j", so:

- σ_{XX} is "direct" X stress, that is stress in the X direction on a plane of constant X.
- τ_{YZ} is "shear" YZ stress, that is stress in the Y direction on a plane of constant Z.

When a stress tensor is described as symmetric this is because opposite off-diagonal terms are the same, that is:

$$\tau_{XY} == \tau_{YX},\ \tau_{YZ} == \tau_{ZY},\ \tau_{XZ} == \tau_{ZX}$$

Rotating a tensor to give element local stresses

If we have a set of direction cosines in the 3x3 matrix `[R]` then we can rotate a tensor thus:

$$[S'] = [R][S][R]^T \quad \text{where } [S'] = \begin{bmatrix} \sigma'_{xx} & & \\ \tau'_{yx} & \sigma'_{yy} & \\ \tau'_{zx} & \tau'_{zy} & \sigma'_{zz} \end{bmatrix}$$

This is how the global to local transformation of stresses and strains is carried out when the **FRAME_OF_REFERENCE** is set to **LOCAL** : the `[R]` matrix is formed from the local axes of the element. The prime " ' " notation is used to signify that the component is in the local (as opposed to global) coordinate system.

Computing von Mises Stress: The deviatoric stress

This is given by:
$$1/\sqrt{2}\ [(\sigma_{XX}-\sigma_{YY})^2 + (\sigma_{YY}-\sigma_{ZZ})^2 + (\sigma_{ZZ}-\sigma_{XX})^2 + 6(\tau_{XY}^2 + \tau_{YZ}^2 + \tau_{XZ}^2)]^{1/2}$$

Computing signed von Mises Stress

This is given by: SGN * $\sigma_{\mathrm{vonMises}}$

SGN is calculated either as the sign of the principal stress with the greatest magnitude or as the sign of the first stress invariant (I1 = $\sigma_{xx} + \sigma_{yy} + \sigma_{zz}$). The method is selected in the preferences file (see Appendix B).

Computing PRESSURE: The hydrostatic pressure

This is given by: $-(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) / 3.0.$ (Note: compression +ive.)

Computing PRINCIPAL stresses

The principal stresses (maximum, middle, minimum) are the three roots (P) of the cubic:

P_{MAX}, P_{MID}, P_{MIN} solve
$$P^3 - (\sigma_{XX} + \sigma_{YY} + \sigma_{ZZ})\ .\ P^2 + (\sigma_{XX}\sigma_{YY} + \sigma_{YY}\sigma_{ZZ} + \sigma_{ZZ}\sigma_{XX} - \tau_{XY}^2 - \tau_{YZ}^2 - \tau_{ZX}^2)\ .\ P - (\sigma_{XX}\sigma_{YY}\sigma_{ZZ} + 2\tau_{XY}\tau_{YZ}\tau_{XZ} - \sigma_{XX}\tau_{YZ}^2 - \sigma_{YY}\tau_{XZ}^2 - \sigma_{ZZ}\tau_{XY}^2) = 0$$

where $P_{MAX} > P_{MID} > P_{MIN}$.

Computing MAX_SHEAR_STRESS

This is given by: $(P_{MAX} - P_{MIN}) / 2.0.$

Computing DEVIATORIC PRINCIPAL stresses

Deviatoric principal stresses (**DEV_PRINC_STRESS**) are given as the deviation from the hydrostatic pressure, (recall compression is +ive).

They are given by: **[P $_{MAX}$ + PRESSURE], [P $_{MID}$ + PRESSURE] , and [P $_{MIN}$ + PRESSURE].**

Computing 2D PRINCIPAL stresses

2D in-plane principal stresses are computed for shell, thick shell, SPH and DES elements from the element local stresses.

They are given by:

$$P^{2D}_{\max} = \frac{\sigma_{x'x'} + \sigma_{y'y'}}{2} + \sqrt{\frac{(\sigma_{x'x'} - \sigma_{y'y'})^2}{2} + \tau^2_{x'y'}},$$

$$P^{2D}_{\min} = \frac{\sigma_{x'x'} + \sigma_{y'y'}}{2} - \sqrt{\frac{(\sigma_{x'x'} - \sigma_{y'y'})^2}{2} + \tau^2_{x'y'}}.$$

Equivalently, these are the two roots (P) of the quadratic equation:

$$P^2 - (\sigma_{x'x'} + \sigma_{y'y'})P + (\sigma_{x'x'}\sigma_{y'y'} - \tau^2_{x'y'}) = 0.$$

Note!! The in-plane computation ignores any out of plane stresses, for thin shells that is $\tau_{Y'Z'}$, $\tau_{X'Z}$' If these are significant the in-plane principal stresses do not represent the true stress state in the element, so these data components should only be used in plane stress situations.

Computing TRIAXIALITY

Triaxiality is the ratio of hydrostatic pressure and von Mises stress.

This is given by: **- PRESSURE / VON MISES.**

Computing LODE ANGLE

This is given by:

$$tan^{-1}\left\{\frac{1}{\sqrt{3}}\left[2\left(\frac{P_{mid} - P_{min}}{P_{max} - P_{min}}\right) - 1\right]\right\}$$

Computing LODE PARAMETER

This is given by:

$$\frac{(2\sigma_{mid} - \sigma_{max} - \sigma_{min})}{\sigma_{max} - \sigma_{min}}$$

Computing ALTERNATIVE LODE PARAMETER

This is given by: $$\left(\frac{27}{2}\right)\left(\frac{J_3}{\sigma_{eq}^3}\right)$$

Computing SHELL TENSION/COMPRESSION

This component shows whether a shell element is in tension, compression, zero stress or not computed on both sides of the shell. A shell in bending will show compression (blue) on one side and tension (red) on the other side. The contour colour will depend on which side of the shell is currently visible.

A surface is defined as being in tension if the magnitude of the maximum principal stress is greater than the magnitude of the minimum principal stress and vice versa for compression. The value on a surface may not be computed if, for example, the top surface integration point is not included in the results.

Computing YIELD_UTILISATION_FACTOR

This is given by: $$\frac{VonMisesStress}{YieldStress}$$

Computing YIELD_UTILISATION_PERCENTAGE

This is given by: $$\frac{VonMisesStress}{YieldStress} * 100$$

16.24.2. Manipulations of Strain Tensor Components

Manipulations of strain tensor components

We have adopted the notation for global strains:

$$[E] = \begin{matrix} [\epsilon_{xx} & &] \\ [\epsilon_{yx} & \epsilon_{yy} &] \\ [\epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz}] \end{matrix}$$

(Symmetric: Upper triangle = lower triangle.)

Where ϵ_{XX}, ϵ_{YY}, ϵ_{ZZ} are the "direct" strains, and ϵ_{XY}, ϵ_{YZ}, ϵ_{XZ} the shear strains.

Important Note: shear strain tensor terms

In many engineering textbooks the strain component is multiplied by a factor of two (for example see "Voigt" notation):

$$\gamma_{xy} = 2\epsilon_{xy}, \quad \gamma_{yz} = 2\epsilon_{yz}, \quad \gamma_{zx} = 2\epsilon_{zx}.$$

This is because some equations can be simplified, for example where

$$\begin{aligned} \epsilon_{xy}\tau_{xy} + \epsilon_{yx}\tau_{yx} &= 2\epsilon_{xy}\tau_{xy} \quad \text{(by symmetry)} \\ &= \gamma_{xy}\tau_{xy} \end{aligned}$$

$\gamma_{xy}, \gamma_{xy}, \gamma_{xy}$ are often referred to as "Engineering Shear Strain". (This is not to be confused with "Engineering Strain".)

In conclusion:

- The shear strain terms written by LS-DYNA in the strain tensor are "true" strains: ϵ_{XY}, ϵ_{YZ}, ϵ_{XZ}
- The shear strain terms used in the formulae on this page are the same.
- There is ***absolutely no factor of 2 involved*** !

For Nastran OP2 results:

- Nastran outputs shear strain terms with a factor of 2, however for consistency D3PLOT applies a factor of 0.5 when they are read.
- If you want to plot the "Engineering Shear Strains" you can use the **GXY_GAMMA_XY_STRAIN** , **GYZ_GAMMA_YZ_STRAIN** and **GZX_GAMMA_ZX_STRAIN** components which are the "true" shear strain components multiplied by 2.

Computing equivalent strain values

These are all computed in exactly the same way as the stress terms above, substituting `[E]` for `[S]` , ***except*** that the von Mises strain has a factor of 2/3 applied:

This is given by:

$$2/3\sqrt{2} * [(\epsilon_X-\epsilon_Y)^2 + (\epsilon_Y-\epsilon_Z)^2 + (\epsilon_Z-\epsilon_X)^2 + 6(\epsilon_{XY}^2 + \epsilon_{YZ}^2 + \epsilon_{XZ}^2)]^{1/2}$$

The reasons for applying this factor of 2/3, which make the calculation suitable for the plastic regime, are given below under " What is von Mises strain? "

Note, **SAV_AVERAGE_STRAIN** is the strain equivalent to **PRESSURE** .

Computing PEMAG_PLAST_STRN_MAG

This is an averaged scalar term reflecting the maximum plastic strain related to the inelastic strain tensor at a given location.

It is given by: $\epsilon^P = \left\{ 2/3\, \epsilon_{ij}^P \epsilon_{ij}^P \right\}^{0.5}$ where ϵ_{ij}^P is the plastic strain component.

Or, in full, the elastic strain tensor is derived from the stress tensor:

$$\epsilon_{xx}^E = \frac{\sigma_x - \nu\sigma_y - \nu\sigma_z}{E}, \quad \epsilon_{yy}^E = \frac{\sigma_y - \nu\sigma_x - \nu\sigma_z}{E}, \quad \epsilon_{zz}^E = \frac{\sigma_z - \nu\sigma_x - \nu\sigma_y}{E}$$

and $\epsilon_{xy}^E = \frac{\sigma_{xy}}{2G}, \quad \epsilon_{yz}^E = \frac{\sigma_{yz}}{2G}, \quad \epsilon_{zx}^E = \frac{\sigma_{zx}}{2G}.$

where ϵ_{ij}^E is the elastic strain component,

ν is Poisson's ratio,

E is Young's modulus, and

$G = \frac{E}{2(1+\nu)}$ is the shear modulus.

The plastic strain tensor is given by: $\epsilon_{ij}^P = \epsilon_{ij} - \epsilon_{ij}^E.$

Plastic strain magnitude is:

$$\epsilon^P = \sqrt{2/3 \left\{ (\epsilon_{xx}^P)^2 + (\epsilon_{yy}^P)^2 + (\epsilon_{zz}^P)^2 + 2\left[(\epsilon_{xy}^P)^2 + (\epsilon_{yz}^P)^2 + (\epsilon_{xz}^P)^2 \right] \right\}}.$$

Note, often the formula is written using *engineering shear strain* terms, which gives a factor of 0.5:

$$\epsilon^P = \sqrt{2/3 \left\{ (\epsilon_{xx}^P)^2 + (\epsilon_{yy}^P)^2 + (\epsilon_{zz}^P)^2 + 0.5\left[(\gamma_{xy}^P)^2 + (\gamma_{yz}^P)^2 + (\gamma_{xz}^P)^2 \right] \right\}}.$$

For thin and thick shell elements the PEMAG data component should be checked very carefully. Depending on the element formulation and the value of ISTUPD on *CONTROL_SHELL the through thickness (local Z direction) strain may or may not be calculated and written to the strain tensor. If this strain value is omitted PEMAG will tend to underestimate the full extent of the strain in the element.

If you wish to use this data component for shells, especially thick shells, it is strongly recommended that you set up a single element "tensile test" model which stretches a representative element well into the plastic regime. Inspect thickness change and strain tensor to ensure that you see thinning of the shell and the corresponding through thickness strain. Another good test is that for uniaxial loading of this sort the

PEMAG value should be exactly the same as the effective plastic strain as reported by LS-DYNA.

Computing ENGINEERING STRAIN

For shell elements, as well as 2D principal max and min strain (also referred to as true major and minor strain), D3PLOT derives engineering major and minor strain.

This is given by:

$$E_{\text{major}}^{\text{eng}} = 100 \times (e^{E_{\text{major}}^{\text{true}}} - 1.0),$$

$$E_{\text{minor}}^{\text{eng}} = 100 \times (e^{E_{\text{minor}}^{\text{true}}} - 1.0),$$

where $E_{\text{major}}^{\text{true}}$, $E_{\text{minor}}^{\text{true}}$ are 2D principal max and min strains (see Computing 2D PRINCIPAL stresses).

For shell elements, D3Plot also derives engineering through-thickness strain.

This is given by: $100 \times (e^{\epsilon_{z'}} - 1.0)$

where $\epsilon_{z'}$ is the through thickness strain, that is, the strain tensor component in the local Z direction. This can be non-zero to preserve volume of the shell.

Computing INTERNAL ENERGY DENSITY for solids

D3PLOT can derive the internal energy density for solids in the elastic regime. This is not computed by default, users wishing to use it should contact Oasys Ltd. The equation implicitly assumes elastic strain, hence there is a factor of 0.5 as the area under the elastic stress/strain curve is a triangle. In the plastic regime this factor needs to be closer to one.

This is given by:

$$\frac{1}{2}\sigma_{ij}\epsilon_{ij} = \frac{1}{2}\left(\sigma_x\epsilon_x + \sigma_y\epsilon_y + \sigma_z\epsilon_z + 2(\tau_{xy}\epsilon_{xy} + \tau_{xz}\epsilon_{xz} + \tau_{yz}\epsilon_{yz})\right)$$

16.24.3. Further Explanation of Strain Components

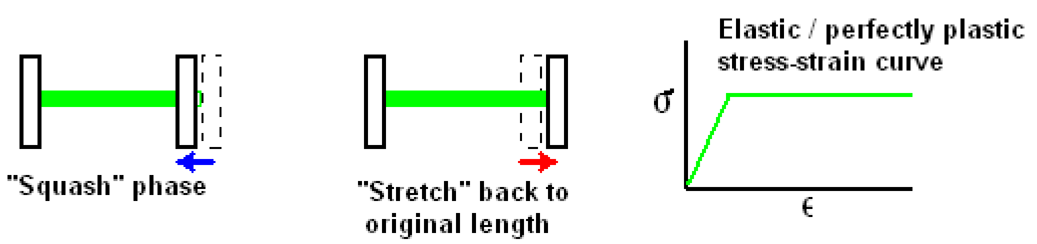
Further explanation of Strain components

LS-DYNA writes out two sorts of strain values:

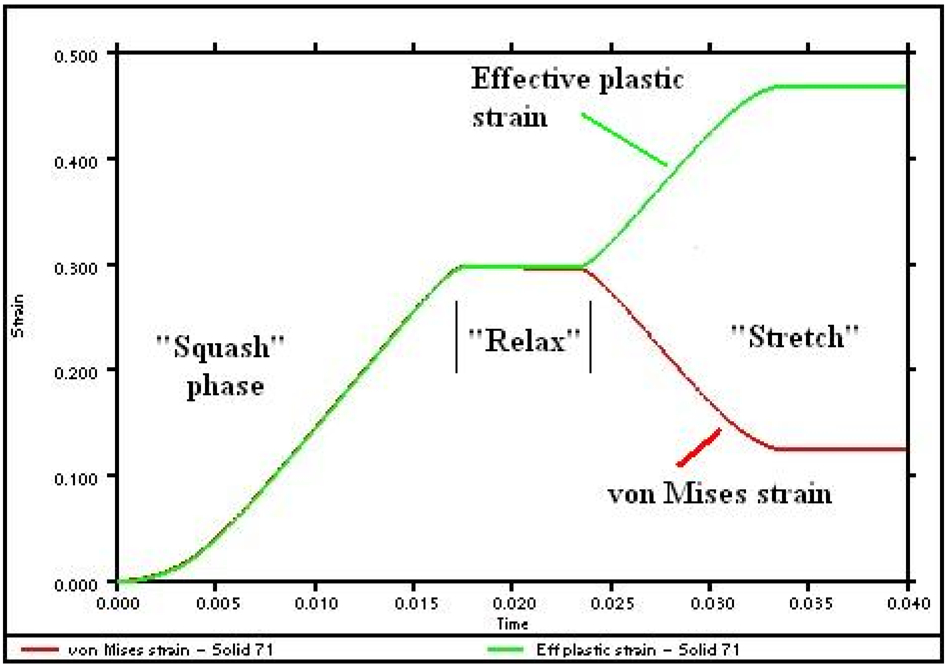
Effective Plastic Strain **(written at each integration point)**	This is a scalar value which is the sum of all plastic strain increments in this element to date. • It is directionless • It is always +ve • Its value will only ever increase • It does not include any contributions from elastic strains It is a measure of the cumulative plastic deformation undergone by this element.
Strain Tensor **(At all intg points in solids, but top and bottom intg points only in shell elements)**	This is a full tensor [Exx, Eyy, Ezz, Exy, Eyz, Ezx] describing the current state of strain in the element. • Being a fully populated tensor it contains directional information • Values can be +ve or -ve • Values can both increase and decrease • It contains both elastic and plastic strain contributions. It is a measure of the current (instantaneous) state of strain in the element.

The question is often asked "how are these two values related?" Perhaps the easiest way to understand this is to consider a simple tensile test specimen involving an elastic/plastic material in which the specimen is first squashed into the plastic regime, then stretched again to its original shape.

"Tensile test" specimen: first squashed, then stretched to original length



The graph below shows curves of Effective Plastic strain (green) and von Mises strain (red) derived from the strain tensor for a typical element at the centre of the specimen:



Observe how in the squashing phase the two are identical, but that when stretching (implying a reversal of direction) takes place then:

- effective plastic strain continues to increase, since the material is still undergoing plastic deformation, albeit in the opposite direction.
- von Mises strain reduces by a similar amount, since the element's shape is reverting back towards its unsquashed condition.

(In an ideal world the von Mises strain curve in this analysis would drop back to the linear elastic value only as the specimen is pulled back out to its original length, but in practice the specimen in this analysis was allowed to bulge and hourglass a bit, meaning that it did not return to the original undeformed shape. This is why both strain curves level off at about 0.034s..)

Perhaps a good way to think of this is that the strain tensor contains information about the current element shape (or, more precisely, the distortions required to get from the original shape to the current one); and the effective plastic strain contains historical information about all the permanent deformation increments required to achieve that shape.

When loading is in a single direction then the effective plastic and von Mises strains will be more or less the same, as in the uniaxial "squash" phase above. (The von Mises strain contains elastic as well as plastic values, so it may be slightly greater.) However as soon as the loading direction, or more precisely the direction of deformation, changes then the two will diverge.

What is von Mises Strain, and why does it have a factor of 2/3?

Von Mises strain, sometimes referred to as "Equivalent strain", is the equivalent of von Mises stress in that it measures the deviatoric component of strain, and

VM stress / VM strain = 3G

and

G = **E / 2(1 + υ)**

where **G** is the shear modulus, **E** is the Young's modulus and **υ** is the Poisson's ratio.

Note that VM stress / VM strain does ***not*** equal the Young's modulus **E** except in the fully plastic state. To understand why consider the following:

Poisson's ratio (**υ**) only lies in the range 0.0 - 0.5 for ***elastic*** strain calculation, once the material goes ***plastic*** then the relationship between longitudinal and transverse strains is controlled by volume conservation as the material distorts permanently, and the plastic Poisson's ratio is required to be 0.5 in all cases if volume is to be preserved. Therefore for "conventional" materials that preserve volume, which implicitly excludes foams and other strange materials, we can eliminate the variability of Poisson's ratio from the problem allowing us to calculate a von Mises strain that has a fixed relationship with von Mises stress in the plastic regime.

In the elastic regime, where the Poisson's ratio is less than 0.5:

3G is less than **E** since **E = 2(1 + υ) G** therefore VM stress / VM strain gives some value < **E**

In the fully plastic regime where the Poisson's ratio = 0.5:

3G = **E** since **E = 2(1 + 0.5) G** therefore VM stress / VM strain should give - more or less - the current plastic modulus value in the element.

To see where factor of 2/3 comes from consider the following:

Perform the von Mises calculation for strain derived from stress with Poisson's ratio set to 0.5 and the strain comes out 1.5x larger. To demonstrate this consider the case of uniaxial stress in the plastic regime:

Let X stress $\boldsymbol{\sigma_x}$ = 1.0, all other stresses are zero.

Let the notional plastic modulus **Ep** be 1.0 for simplicity, then using 0.5 for Poisson's ratio we obtain the following strains. Stresses are shown for comparison.

Strain values **Stress values**

X strain $\boldsymbol{\varepsilon_x = 1.0}$. $\boldsymbol{\sigma_x = 1.0}$

Y strain $\boldsymbol{\varepsilon_y}$ = -$\boldsymbol{\upsilon}$. $\boldsymbol{\sigma_x}$ / **Ep** = **-0.5** $\boldsymbol{\sigma_y = 0.0}$

Z strain $\boldsymbol{\varepsilon_z}$ = $\boldsymbol{\varepsilon_y = -0.5}$ $\boldsymbol{\sigma_z = 0.0}$

Performing the conventional von Mises calculation used for stress we obtain:

von Mises strain **= 1/sqrt(2).((**$\boldsymbol{\varepsilon_x - \varepsilon_y}$**)**2 + (**$\boldsymbol{\varepsilon_y - \varepsilon_z}$**)**2 + (**$\boldsymbol{\varepsilon_z - \varepsilon_x}$**)**2) = 1.5**

Whereas we can see by inspection that the von Mises stress **= 1.0**

Therefore a factor of 2/3 factor is included in the formula for von Mises strain so that, during the plastic deformation phase of a uniaxial tensile test on a metal, VM strain will be the same as uniaxial strain, and therefore the plastic stress-strain curve from a tensile test can be used as input to material models expecting VM stress versus VM strain (after conversion from nominal stress/strain to true stress/strain).

But what does von Mises strain actually mean?

This section is being written because this question has been asked so many times. To be truthful it is not very useful!

- In predominantly plastic analyses it can be thought of as the quantity on the strain axis of the material's stress-strain input curve. When strains are occurring in two or three directions, the Von Mises strain gives the strain in a uniaxial tensile test that would work-harden the material to the same degree.
- In predominantly elastic analyses it has limited usefulness.

It is normally the case that Effective Plastic strain is far more useful in fully plastic analyses, and that the individual (directional) and principal components of the strain tensor are of more interest in elastic ones.

17. D3PLOT Use of Graphics Hardware

D3PLOT USE OF GRAPHICS HARDWARE

D3PLOT supports 2 categories of graphics devices:

X_Windows	`X(option)`	2-D windows on any Unix hardware, and on PCs via emulation.
OpenGL	`OPENGL`	3-D windows on all common hardware. Only available method under Windows.

This section gives more details about these device categories, and describes what capabilities are available on each one. You should not normally need to read this section, and it is included for interest only. If you have problems with graphics hardware please contact Oasys Ltd for advice.

17.1. The "X" (X_Windows) 2-D Protocol

The "X" (X_Windows) 2-D protocol

X_Windows is the most widely supported graphics protocol on modern engineering work-stations. It has the following attributes:

> It uses a client/server mode of working that makes it network transparent. That is you can display results on one machine (the server), while actually running on another, (the client). Client and server may be totally different machines connected by a network, or indeed the same machine working autonomously.
>
> It supports a wide range of screen types. Almost any graphics screen of any resolution and type will work under this protocol.
>
> It integrates well with window managers. All current workstation window managers will support multiple "X" windows.
>
> It is two dimensional only. (The PEX 3D extension has proved unsuccessful in the face of competition from OpenGL.)

The D3PLOT screen menu is an X-Window, a child of the window manager, and it has its own children that it manages locally.

The menu system always uses the default visual of the screen so that, in itself, it will not clash with other applications. The graphics sub-window within this menu may also be invoked with the default visual, or with a different one.

X "visuals" are discussed in the next section.

17.1.1. X_Windows Colour Visuals and their Attributes

X_Windows colour visuals and their attributes

Since the X_Windows protocol is designed to run on a wide range of hardware it offers a range of four colour "visuals" to employ the various graphics screens to best effect.

VISUAL TYPE	Typical #Bit-planes	#Colours available	Colourmap type
PseudoColor	4 - 12	16 - 4096	Read/write
StaticColor	4 - 12	16 - 4096	Read-only
DirectColor	4 - 24	16 - 16777216	Read/write
TrueColor	4 - 24	16 - 16777216	Read-only

You don't have to understand X visuals fully to use D3PLOT, but you should be aware that the visual type and number of bit-planes you use have an influence on image quality, interaction between the various windows on the screen and animation speed.

#Bit-planes:	The number of colours available is $2^{\#BIT-PLANES}$. Most display modes in D3PLOT will function with 16 colours, ie 4 bit-planes; but the two lighting modes **SH** **(GREYSCALE)** and **SI** **SHADED_IMAGE** need at least 100 colours to give decent results, and work best with 256 or more colours. So using more bit-planes will give better quality images: on a 24 bit-plane visual D3PLOT will give "true" colour rendering. However animation requires images in memory (pixmaps) to be transferred to the screen, and the greater the number of bit-planes in an image the longer this takes. So using more bit-planes gives slower animation.
Map type:	Read/write colourmap visuals (Pseudocolor and Directcolor) permit colours to be changed dynamically on the screen without re-drawing, whereas read-only ones require the image to be re-drawn if colours are to be changed. In addition read/write colourmaps permit entries to be created that match as exactly as possible the shades required, whereas read-only maps have to select shades from those available: this is not a problem on 24 bit-plane screens where all possible shades exist, but it can give inferior lighting plots on devices with fewer bit-planes.

More significantly read/write visuals generally require multiple screen windows to share colourmaps, and this can lead to colours in some windows changing as entries which are correct for one window conflict with those for another. Read-only maps never suffer from this problem.

17.1.2. Choosing an X_Windows Visual

Choosing an X_Windows visual

From the above discussion it should be clear that there is no "best" X_Windows visual for D3PLOT, and indeed a high performance 24 bit-plane visual may be a liability if fast animation is required. Therefore D3PLOT release 8.0 allows you to choose any visual supported by your server. When it asks you to give a device type:

You may select one of the following X options:

X8	Using an 8 bit-plane visual for fast animation. D3PLOT will use a TrueColor visual if it exists, failing that then PseudoColor and finally StaticColor. Shading and lighting should be acceptable, but dithering will be required to reproduce all shades.
X24	Using a 24 bit-plane visual for best quality lighting and shading. D3PLOT will choose a TrueColor or DirectColor visual. Shaded and SI plots will be of good quality, but animation will be slower as three times as much information has to be moved around.
XMENU	Interactive selection of visuals from menu. This option presents you with a menu of the visuals available on the server and lets you select one. Considerable on-line help is available.

X8 will be adequate for most users, **X24** is only justified if you are going to be grabbing screen images and need the higher image quality.

Note that "dithered" laser plots are sent to the screen at the depth of the visual, but are always generated in "true" (effectively 24 bit-plane) colour in the laser file.

17.2. 3D Protocol: OpenGL

3D protocol: OpenGL

The OpenGL graphics library has become the de-facto industry standard, and is available on virtually all hardware.

OpenGL itself has no native window support, it runs under the windowing system of its host machine. Therefore on Unix machines OpenGL rendering takes place within an X window, and on PCs within a "windows" window. In environments that support it

OpenGL can operate in separate client/server mode, and using "objects" in the server can be efficient in this mode.

OpenGL is similar in most respects to X_Windows except that it is fully three-dimensional and utilises hardware acceleration for most graphics functions. In particular:

> The image can be rotated, translated and scaled dynamically using the mouse as under X. But this is now carried out by the hardware so, depending on hardware power and size of model, this can be done far faster.
>
> All calculations involving hidden-surface removal are done by the hardware using Z-buffering, generally far faster than they can be achieved in software.
>
> Likewise all shading and lighting calculations are done in hardware, again far faster than in software. Smooth (gouraud) shading and transparency are implemented in hardware.
>
> "Clipping", the ability to calculate the intersection between the image and arbitrary planes, is provided. This provides facilities such as "Z-clipping" which are not available in software.

OpenGL will give better performance than 2D X under nearly all circumstances. The exceptions to this are:

Large animations:	3D animations require up to4 times the amount of memory as the equivalent under X. If you attempt to create large animations you may run your machine out of memory, or at least cause it to "page" unacceptably. Very large (ie product of #vectors x #frames) animations are best carried out under X for this reason.
Hidden-line plots:	Hidden-line (and line contour) plots produced by Z-buffering are not as good as those generated in software. You can get round this by switching temporarily back to 2-D mode.

The complications of different visuals and numbers of bit-planes do not usually apply to OpenGL. They operate in "RGB" mode (roughly equivalent to Truecolor), and generate intermediate shades by hardware dithering if not enough bit-planes are available to produce the required colour directly.

If your device supports 3D you should use it.

17.3. Summary of Capabilities of Each Graphics Protocol

Summary of capabilities of each graphics protocol

The following table summarises the capabilities of the various graphics devices listed in this section.

FUNCTION	X-Windows	OpenGL
Display mode:		
LI (**LINE**)	x	x
HI (**HIDDEN_LINE**)	x	x (1)
LC (**LINE_CONTOUR**)	x	x (1)
CT (**CONTINUOUS_TONE**)	x	x
VE (**VECTOR**)	x	x
ARROW	x	x
SH (**GREYSCALE**)	x (2)	x
SI (**SHADED_IMAGE**)	x (2)	x
Function:		
Animations	x	x (3)
Plots may be stopped	x	x
Dynamic viewing	x	x

Notes:

(1)

Hidden-line and line contour quality under 3D is adequate. Better results may be obtained by switching temporarily to 2D mode.

(2)

Quality on visuals with fewer than 8 bit-planes not good.

(3)

Very large animations under 3D may run the computer out of memory. Switching to 2D mode may be better for these.

Some minor differences between output format on different devices may be found, for example character sizes, but generally plots should look the same on all devices.

18. Problem Solving

PROBLEM SOLVING

This section describes some common problems, and gives suggestions about solving them. It doesn't attempt to cover all possible causes: please call Oasys Ltd if you cannot solve your problems.

18.1. Problems Reading Files

Problems reading files

The files are read but seem to be corrupt on initialisation

If the node and element numbering is reported as being scrambled it is likely that the analysis job crashed on initialisation or, if the job is still running, it may not yet have initialised. Check to see if it crashed (see the **.LOG** file). If it is still running try reading it again later when it has got a bit further (this is because the node and element arbitrary numbering tables are the last items to be written at job initialisation).

Alternatively the automatic file format detection may have mis-diagnosed the file's data format. Check that the **FILE_TYPE_32** or **_64** environment variables are set correctly ([Disk format of binary database files](#)). This is particularly the case when reading older 64 bit files.

D3PLOT crashes when reading in a file

This generally means that the **.PTF** file is corrupt. For a job that is still running wait for it to be initialised, for a job that has finished this means that it crashed during initialisation leaving an incomplete file. The D3PLOT crash is due to trying to work out coordinate limits from nonsensical values.

The crash can also be caused by trying to read 64 bit files in **IEEE** rather than **Cray** format, or vice-versa. See [Disk format of binary database files](#)

The last state(s) in the file seem to have nonsensical times and/or corrupt results

This usually means that a complete state has been only partially written. If the analysis job is still running it usually means that the computer system buffers are still holding data waiting to be written to disk. Wait to see what happens when the next state is written as this will probably sort out its predecessor. You can force a dump of a plot state by using the `STATUS` command in the command shell.

If the job has finished then it probably crashed or ran out of disk space. Check the `messag` or `.otf` files for error messages, check the `.log` file for evidence of crashes or disk space exhaustion.

`D3PLOT reports that the .XTF or .CTF files are incompatible with the .PTF file`

This means that the control parameters in the various files suggest that they have come from different analyses. The most common cause of this is that files from an old analysis of the same name have not been deleted when the new one is run: check the creation dates on the files.

D3PLOT will ignore the incompatible `.ctf / .xtf` file(s) and continue running.

D3PLOT issues warning that >1000 element meet at a node

This invariably means that the file is either corrupt (ie completed analysis job crashed on initialisation), or is still waiting to initialise (job still running). Take action as described for when D3PLOT crashes reading a file.

D3PLOT issues warning that solid or thick shell elements have crossed faces

This may not be an error: it is possible (but unusual) for this situation to arise legitimately. If this is the only message issued and D3PLOT initialises normally have a look at the offending elements and check their topology. If this is blatantly wrong it means that the file is corrupt.

D3PLOT warns that 6-noded thick shells have mis-numbered faces

It is a common error to create triangular thick shells using the solid "wedge" numbering sequence (1,2,3,4,5,5,6,6) instead of the "extruded triangle" sequence (1,2,3,3,4,5,6,6). LS-DYNA will still run, but the results for these elements will be dodgy. D3PLOT will also run, but the face numbering for these elements will be wrong leading to mis-diagnosis of external faces, free edges and data averaging at nodes.

D3PLOT issues warning that duplicate solids or thick shells exist

Again this may indeed be the case. Check as described above, and take heed that coincident solids may not be displayed unless the **DISPLAY_OPTIONS** , **INTERNAL_FACES** switch is turned **on** . (Since all their faces will be marked internal.)

`Problems with missing .PTF file family members: last <n> states not read`

If the final <n> states from your analysis are not read, and these appear to be in the last family member(s), check for gaps in family member numbering. If one or more family members appear to be missing check the reason. However you can skip gaps in the family member sequence using the **FILE >** , **FILE_SKIP** command: see The FILE > popup menu options .

D3PLOT fails to read some states from a file, with a regular pattern

This can occur if the family member size has been set to a value smaller than that used when writing the files, as states in the tail end of family members are not read.

In D3PLOT 8.0 onwards family member size detection should be automatic, but you can over-ride it using the **FILE >** , **FAM_SIZE** command, or by setting the **`FAM_SIZE`** environment variable as described in The FILE > popup menu options .

File protection problems

D3PLOT does not require "write" access to database files, so you can process results to which you only have "read-only" access.

However operations that create files: laser plotting, view storage and session file generation, all require "write" access to the relevant directories. If a file open fails due to protection errors you will be warned and the operation will be aborted.

18.2. General Graphics Problems

General graphics problems

The terminal won't draw anything

There are several possible causes. Have you:

- **BLANKED** everything out of sight? Try turning the **BLANKING** switch off.
- **CLIPPED** everything out of sight? Try turning the **VOLUME_CLIPPING** switch off.
- Turned all entities off? Check the status of the **ENTITY** settings.
- Scaled, rotated, zoomed everything out of sight? Try a **ZERO** command.

- Are your deformations so large that the image at this complete state time is off the screen? Try an **AC** command.
- Got a corrupt `.PTF` file? (Job crashed)

Facets of solids and thick-shells disappear when the model is very distorted

This is because the "back face" detection algorithm can get confused when elements become very misshapen, and removes faces that it shouldn't. Try turning the **DISPLAY_OPTIONS** , **BACK_FACE** switch on, which should cure the problem. Alternatively you can reduce distortions artificially by setting the displacement magnification factors to values less than 1.0: see **DEFORM** , **MAGNIFY_DISPLACEMENTS** .

2-D hidden-line removal seems to make mistakes

The default "painter" algorithm is cheap and cheerful, and it can make mistakes. You can prevent this by using the more expensive "rigorous" algorithm: see **DISPLAY_OPTIONS** , **HIDDEN_LINE_OPTIONS** .

Problems displaying stonewalls

Stonewalls, especially infinite ones, present a problem since their characteristic size is so much greater than typical facets in a model. This can show up very clearly in hidden-line plots where they often appear to be in front of or behind where they should be.

This is usually due to their interaction with the perspective calculation, and turning this off (**PERSPECTIVE** , **OFF**), and switching to "rigorous" hidden-line removal will improve matters considerably.

18.3. Memory Consumption Problems

Memory consumption problems

The message unable to obtain more memory is given

This means that an attempt to allocate more memory has failed because the system has refused to give it. This can happen in many different contexts, but is most common when building animations under OpenGL, as this is a memory-intensive process.

The problem may be soluble at the system level:

(1) All systems

- Check that other unnecessary processes competing for system resources have been shut down. Use the **ps** command (Unix) or the Task Manager (Windows NT) to examine system usage.
- Check that you have enough swap space configured. (System administrator privileges are required to alter this.)

(2) Unix systems: Make sure that the operating system is not imposing arbitrary limits.

- Artificial limits may be imposed in your command shell. The **unlimit** command can be used to lift all restrictions that you, as a user, have privileges to change.
- The maximum "data segment size" in the kernel may be set to a low value. Many Unix systems come configured with this set to their physical memory size, which stops a given process spilling far into swap space. You will need system administrator privileges to change this as it requires the kernel to be modified and rebuilt.
- Check that the window manager process has not been running for a long time, and has accumulated a lot of memory. If it has it may be necessary to kill it (ie revert to console mode) then restart it using **startx** .

If these do not work then you will have to reduce the amount of memory you are using within the D3PLOT process itself:

- Use the **FILE >MEMORY** then **VIEW** database command to empty (partially or fully) the database.
- Change the animation mode from **VECTOR** to **DIRECT** (**ANIM >** , **DISPLAY_MODE** : see DISPLAY_MODE The display mode used for graphics) to save memory.
- If you have two computers try using client / remote server mode. See DISPLAY_MODE The display mode used for graphics .
- See Memory Management on "memory management"

18.4. Graphics Problems

Graphics problems

Problems with X_Windows addresses over a network

The X_Windows protocol makes a distinction between the process running the computer programme, the "client", and that displaying the graphics, the "server". This is to enable a software package to operate in the same way whether it is running on the local machine, or on a remote machine over a network.

Where client and server are the same machine then no confusion arises since there is no ambiguity about where graphics should be displayed. However when you are displaying results over a network from a remote client you may get the message:

```
Unable to open display :0
```

This means that the host machine must be told where to display graphics. On UNIX machines you do this by setting the `DISPLAY` environment variable as follows:

C shell:	`setenv DISPLAY <server>:0`
Bourne shell:	`DISPLAY=<server>:0 export DISPLAY`

Where `<server>` is the network address of the machine on which you are displaying the graphics. A raw address (eg `69.60.10.1`) may always be used or, if the host machine knows about your server, you can use the machine's name.

Multiple screens : The " `:0` " is the screen id on that machine, and this will almost always be screen zero as here. The exception is when you have multiple screens attached to a device, in which case the syntax will become " `:0.0` " for the first screen, " `:0.1` " for the second screen, and so on.

Therefore typical destination commands might be:

`setenv DISPLAY snoopy:0`	(Machine "snoopy", screen 0)
`setenv DISPLAY 69.23.15.2:0`	(Address 69.23.15.2, screen 0)
`setenv DISPLAY :0.1`	(This machine, screen 1)

(**nb:** On UNIX machines see the file `/etc/hosts` for host names and internet addresses. You will only be able to refer by name to machines in this file. However you can display on any machine by referring directly to its numeric address: the names in this file are just convenient aliases for this.)

Problems when the server refuses a connection

A further complication may arise due to permission not being granted for that host to connect to your server. This will generate the message:

```
Host is not permitted to connect to server
```

In this case you must also tell your server to accept graphics from the host with the command:

```
xhost + <hostname>
```

If you omit <hostname> then access will be permitted to all hosts. Your system manager can configure your system such that access is permanently permitted to a selected list of hosts, or indeed all hosts. The latter (all hosts) is potential security hole since the inter-client communication permitted by X window managers is a possible back door for hackers. Network users: you have been warned!

Screen-picking selects invisible entities

Because hidden surface removal is performed in the hardware Z-buffer it is difficult for the client D3PLOT process to know what is and is not visible in a hidden or shaded plot. The picking algorithms attempt to work this out, but occasionally they make mistakes.

There is no absolute cure for this, but suggestions are: reject the pick, rotate the image a touch, and try again as this may make the selection of the visible element less ambiguous. Or maybe just try again taking more care to pick the centre of the item you want.

The image disappears altogether

This can happen for a variety of reasons:

Z-clipping	It is possible to rotate, translate or scale an image such that it passes totally outside the screen Z clipping region. Try resetting these planes: (**3D_OPTS** , **Reset Z**). Turn the projection box (**3D_OPTS** , **SHOW_PROJ** switch) on to see the view frustrum and clipping plane locations.
Bad matrix	The hardware rotation matrix can sometimes get corrupted if a very large number of screen rotations give rise to progressive ill-conditioning: (ie direction cosines no longer have unit length). In this case reset the matrix with the command **ZERO** (reset to default viewing state).

There can also be less obscure reasons: see "The terminal won't draw anything" in General graphics problems .

Hidden-line quality is poor

This is a limitation of Z-buffered graphics. The problem arises because hidden-line plots are generated by drawing visible borders around black polygons, and sometimes the Z coordinates of the polygon win, and sometimes those of the line win, resulting in a broken line. The breaks will move as the image is rotated. D3PLOT attempts to get round this by raising the lines slightly above the polygon surface, but this doesn't work very well when polygons are near edge-on to the viewer.

You can adjust the amount by which the polygon borders are lifted above the surface by holding down the < `left control` > button and the < `left mouse button` >. Moving the mouse up the screen will lift lines off the surface towards you, moving it down will move them back down.

Animations become horrendously slow.

Storing animations can use large amounts of memory. This means that you run your machine out of memory much faster and the slowdown is due to it "paging" memory to and from disk.

Unless you are prepared to buy more memory the only thing you can do is to reduce the memory being used by one or more of:

- Reducing #frames and/or #vectors by simplifying the plot.
- Using a "cheaper" display mode.
- Switching to **DIRECT** animation mode - slower, but uses no memory.
- Emptying some of your database memory (**FILE >** , **DATABASE** , **EMPTY_xxx**)

18.5. Miscellaneous Problems

Miscellaneous problems.

Problems with coincident solid or thick-shell elements.

You are warned if coincident solid or thick-shell elements are found when the files are read in. This is because all faces of such elements will be flagged as "internal" and the elements won't be displayed unless you turn the **DISPLAY_OPTIONS** , **INTERNAL_FACES** switch **ON** .

There may also be problems with averaging data at nodes for such elements unless you set the **AVERAGING: MATERIAL_IGNORED** and **BLANKING_IGNORED** switches carefully. See AVERAGING... Controlling data averaging across adjacent elements for more details.

You can list coincident elements of all types with the **WRITE** , **COINCIDENT** command.

Problems displaying extra nodes on rigid bodies

D3PLOT considers nodes that are not attached to elements to be non-structural and does not normally display them. "Extra nodes on rigid bodies" can fall into this category as, unless they are attached to an element, they do not appear to be part of anything. To see these you need to turn the **ENTITY** , **ALL_NODES** switch **ON** .

You can list nodes that D3PLOT considers to be non-structural with the **WRITE** , **UNATTACHED_NODES** command.

Problems seeing contact surfaces.

You can only see your contact surfaces if you have written the (optional) contact force (`.CTF`) file during your analysis. No `.CTF` file: no contact surfaces visible.

If your analysis would take a long time to re-run you can at least visualise the contact surface geometry by reinitialising it under a different name with a contact force file, and then renaming this `.CTF` to look as if it came from the original analysis. D3PLOT will extract the geometry from this but obviously will be unable to read transient contact force data which isn't there. It can cope with this, although you will get the warning message at initialisation that there are no contact force states to match structural states.

Problems seeing springs, joints and stonewalls.

The topology of these elements is extracted from the `.XTF` file so, if it isn't there, you won't see these.

Reinitialising and renaming the `.XTF` file as above will solve this problem.

18.6. MEMORY Viewing and Controlling the Memory Usage for this Process and the Whole Machine

MEMORY Viewing and controlling the memory usage for this process, and the whole machine.

On all operating systems D3PLOT has to co-exist with other processes, and when its consumption of system resources is small this is not an issue. However as you process larger models you will approach the capacity of you machine, and control of memory use will become important. There are two key issues:

Process size, the size of the D3PLOT process, affects performance.

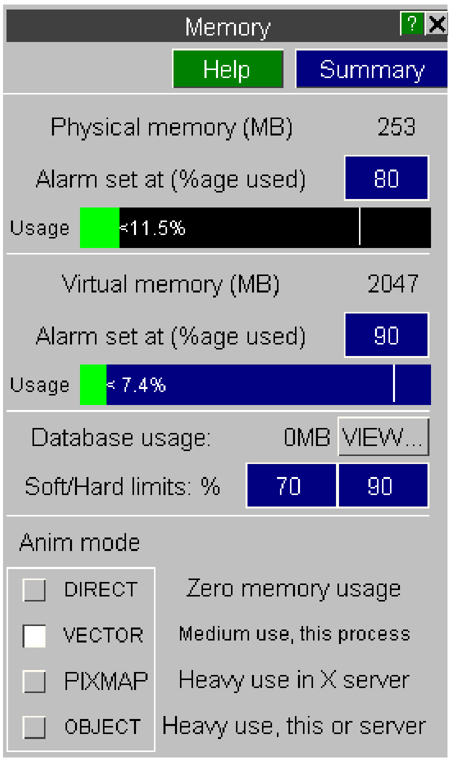
- As this approaches 80 - 90% of the physical memory size of the machine it will become necessary for the operating system to use "virtual" memory. This is because other processes need to maintain a presence in memory and, to make space, some of the D3PLOT process will need to be paged ("swapped out") onto virtual memory on disk.
- Paging, when some of the process pages [(1)] are swapped onto disk, has a major impact on speed because of the delay in reading them back into

memory again when needed. The symptoms are a much slower response, accompanied by the disk chattering away. However the process will continue to run, albeit more slowly.

Virtual memory usage - running out will halt processes, or even crash the machine.

- Operating systems maintain "swap space", which is an area of disk set aside for pages of memory that have been "paged" out of physical memory. This is known as "virtual" memory, and is usually up to 3 times the size of physical memory, but can be any value set by the system administrator.
- Virtual memory allows computers to handle the situation where the sum total of memory space requested by all processes is greater than the physical memory available. They do this by swapping unneeded pages of memory onto disk, then swapping them back into memory (evicting other pages to make space) when they are required again. A "page fault" occurs when an executing process requires a page that is not resident in memory and has to be read in from disk.
- Parking dormant processes on disk has no impact on the performance of running ones. But it will be obvious that if a running process makes excessive use of virtual memory it will generate a lot of page faults, which will slow down its performance.
- If virtual memory space is all used up then the operating system will start to kill processes, and the machine may crash if essential services cannot obtain the memory they need!

The **MEMORY** panel itself is shown here.



This section shows Physical memory statistics. The machine has 253MB of memory. D3PLOT is using 11.5%. The alarm has been set at 80%.

This section shows Virtual memory statistic s. The machine has 2047MB of swap space, of which 7.4% is in use, alarm set at 90%.

This section shows the results database memory usage summary.

Finally this section allows you to alter the graphics display mode. (Some modes use more memory than others.)

Memory usage bars:

The two usage bars (%age physical memory used by D3PLOT , and **total** %age virtual memory usage over the whole machine) are replicated in miniature on the **MEM** button on the front panel for easy reference. So long as they both stay in the green you don't have a problem, and even light orange (60%) is probably OK. More than that and you may need to take action - see below.

Alarm limits:

Both physical and virtual limits have alarm values set at 80 and 90% respectively. If an alarm limit is reached D3PLOT stops what it is doing and maps an alarm panel explaining the problem. You can change alarm values at any time, and set them to large values (eg 1000%) if you want to ignore them altogether.

Database Usage:

This summarises the settings of the results database, and the **VIEW...** button takes you to the main database manager.

Allows you to select an animation mode as described in DISPLAY_MODE The display mode used for graphics . (The same as **ANIM > DISPLAY_MODE**). Some modes use more memory than others.

What to do if memory runs short.

- If virtual memory is short see if there is anything else running on the machine that you can shut down (remember the virtual memory bar is for the whole machine, not just D3PLOT).
- This usually happens when building animations. See if you can cut down what is displayed to save memory:
- Display alternate states instead of every state;
- Use a simpler display mode (eg flat shading, not smooth; **SI** not **CT**)
- Turn off unneeded display items (node symbols, labels)
- If the results database is very full use its **OPTIONS > EMPTY_xxx** commands to release some space, and reduce the soft and hard %ages. (This may not release space back to the operating system, but the space freed can be re-used internally.)
- Use a less expensive display mode:
- **DIRECT** uses no display memory, but is slow.
- **VECTOR** is a reasonable compromise (the default)
- **OBJECT** use a lot of display memory in the server process

In extreme cases you may find that none of the above free enough memory and you have to exit and start again with "cheaper" (in terms of memory) settings. If this happens:

- Before reading in a model use the **MEMORY** button on the filename panel to select the display mode and database soft/hard %ages.
- Then define a filename and read in the model. The new settings will be used ab initio and will result in lower memory usage.

Another problem that can occur following **OBJECT** mode animations is that the graphics server process (owned by the operating system) may grow to a very large size and cause problems. This is because some operating systems don't allow memory to be released once it has been allocated.

It will be necessary to shut down and restart this process in the following way:

- Logout from the display.

- At the "login" prompt choose the "no windows" or "console" mode (usually under an "options" menu). This will shut down the X server completely.
- Most "Common Desktop Environment" (CDE) window managers will automatically restart the X server after a short delay.

Dealing with huge models by using two computers in client / remote server mode.

If, despite everything you do, your model is too big to fit into your machine you may still be able to deal with it by splitting the job over two machines. The "client" machine runs the D3PLOT process, and the remote "server" displays the graphics.

- On the client set the **`DISPLAY`** environment variable to point to the display on the server machine. (Eg **`setenv DISPLAY remote_name:0`**)
- Use **OBJECT** mode this results in the graphics data being displayed in the server, not the client.
- The initial transfer of graphics data from client to server will be slow, since it has to go down a network. But subsequent animation, performed locally on the server, will be fast.
- Keep and eye on the memory usage on the server: D3PLOT does not monitor this for you. (It only "knows" about the client machine.)

DISPLAY_MODE The display mode used for graphics explains how animation works in client/server mode. If D3PLOT Will Not Open a Window on Your Display and Client/Server Graphics Using OpenGL give more information on specifying network addresses.

---

1. A "page" of memory is a conveniently sized quantum, typically 4kB: this varies between systems.

19. REPORTER Integration

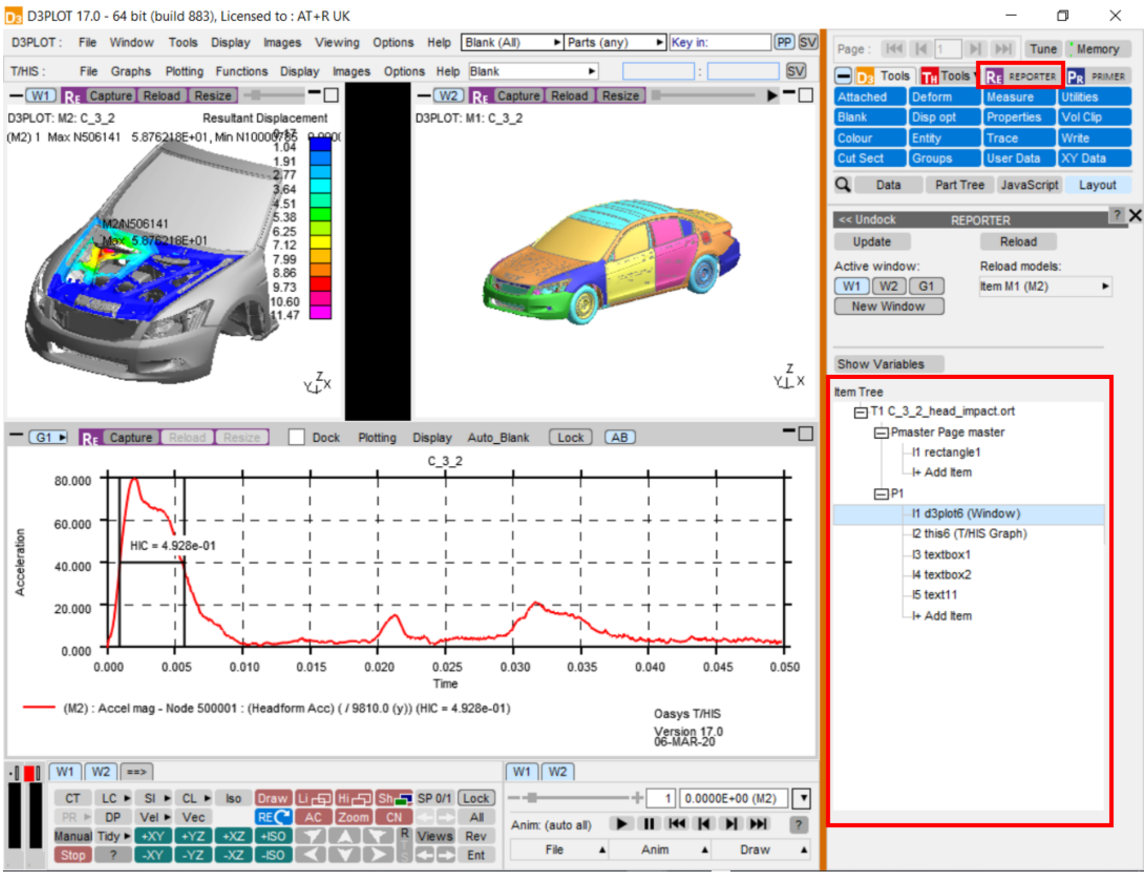
REPORTER INTEGRATION

This section describes how to work with D3PLOT, T/HIS and REPORTER to quickly and easily create reports from results.

19.1. Linking the Programs

Linking the Programs

REPORTER can be opened from D3PLOT and T/HIS using the REPORTER button in the top-right. This opens a linked session of REPORTER, allowing reports to be interactively created and edited. Both D3PLOT and T/HIS can be opened from inside REPORTER too, using the program buttons in the top bar of REPORTER. REPORTER can be connected to both D3PLOT and T/HIS at the same time and the D3PLOT->T/HIS link is also supported. Graphs in T/HIS are treated the same as graphs in a D3PLOT->T/HIS linked session.



19.2. Item Tree

Item Tree

Once a template is opened in REPORTER, all items in the template will appear in the Item Tree in the REPORTER panel in D3PLOT or T/HIS. Selecting an item in the Item Tree will select the corresponding item in REPORTER and vice-versa.

The Item Tree can include items of all types in REPORTER, such as textboxes and images, as well as D3PLOT, T/HIS and PRIMER items. Only placeholders, D3PLOT items and T/HIS items can be overwritten with new D3PLOT or T/HIS items. Placeholder items exist to allow a layout to be created for the report before populating it and can be converted into any other item type.

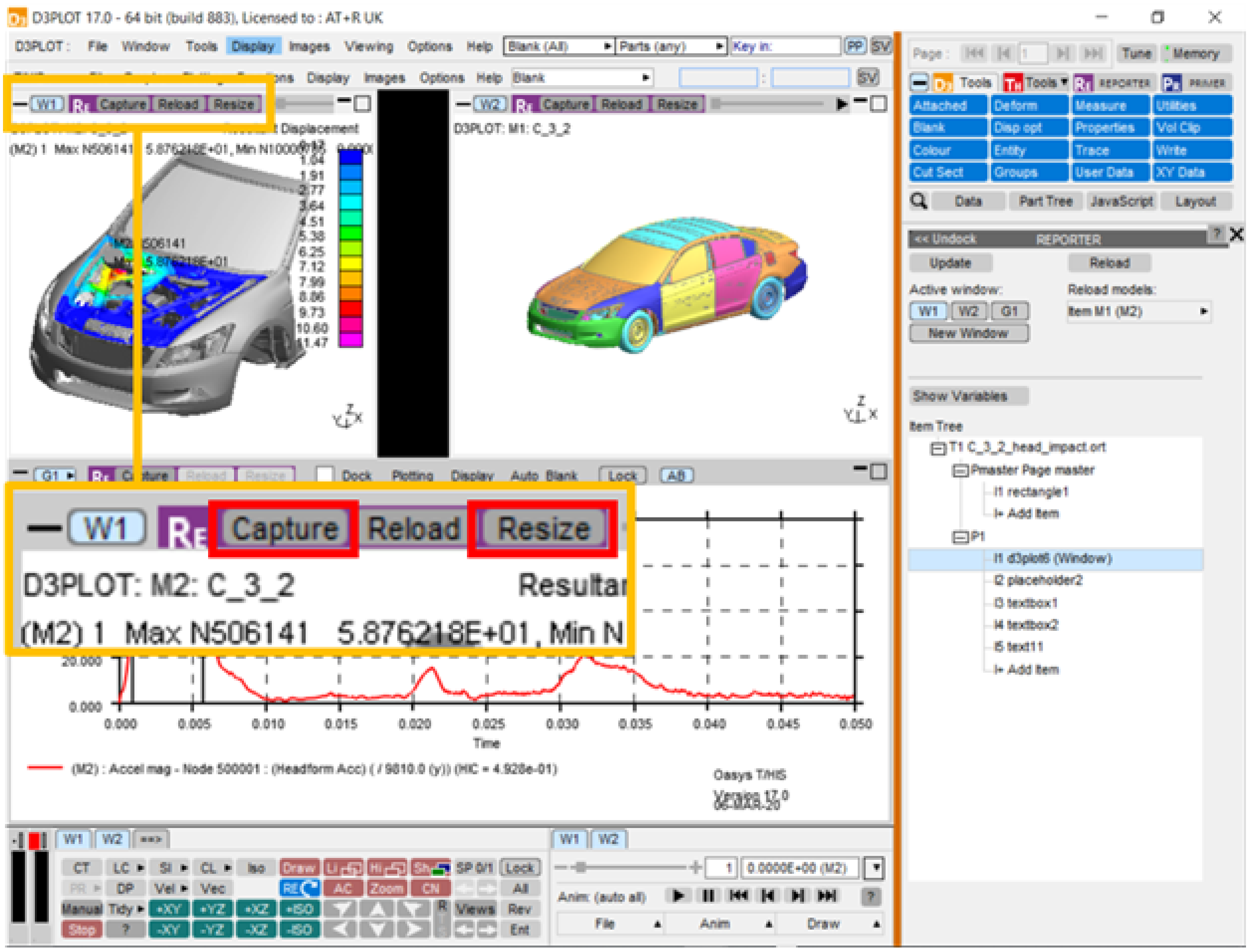
19.3. Capture

Capture

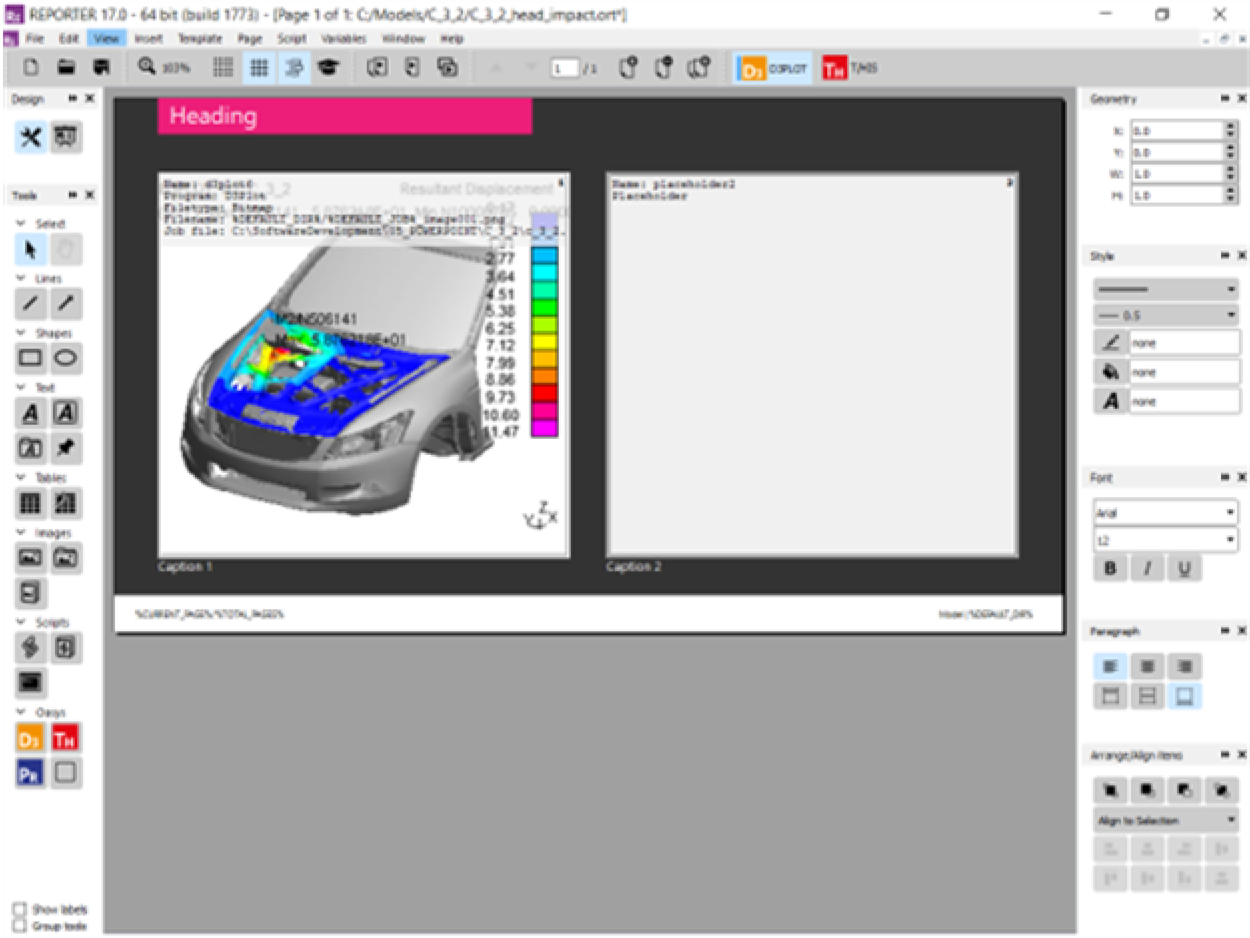
Windows and graphs can be captured into REPORTER, saving an image together with additional information to allow the capture to be reloaded later. For D3PLOT windows, this is a properties and settings file. For T/HIS graphs, this is a FAST-TCF script. Graphs captured in the D3PLOT->T/HIS link are treated exactly the same as graphs in T/HIS, so the resulting items will be identical. Variables containing useful values related to the models or curves in the captured window can be added to the item before capturing (see Variables).

Note that in the Oasys Suite 17.0 method, only single windows and graphs can be captured. The intention being that the windows and graphs are easily captured individually and laid out in REPORTER with greater flexibility.

In order to capture a window, first select the target item in REPORTER, either selecting it directly in REPORTER or using the item tree. You can capture into a new item by selecting 'I+ Add Item' in the item tree. Once the item is selected, the 'Resize' button on the top bar of the window can be used to resize the window to match whatever image size is specified on the selected REPORTER item, such as 'Fit object box'. Finally, either press 'Capture' on the top bar of the target window or select the window in the 'Active window' list in the REPORTER panel and press 'Capture' at the top of the panel.



This will send the information to REPORTER and the image will appear on the item.



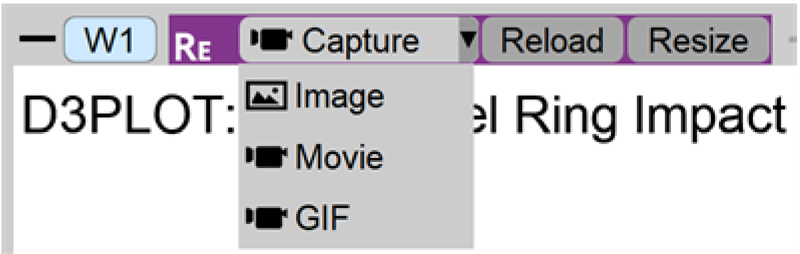
19.3.1. Capturing Movies

Capturing Movies

Starting with D3PLOT 18.0, MP4 movies and animated GIFs can be captured with a D3PLOT Item in REPORTER in place of a static image. The process for Capture is unchanged: just right-click on the Capture button in D3PLOT (either in the REPORTER panel or at the top of the target window) to reveal the new Movie (MP4) and GIF options.

When selecting an existing D3PLOT Item in the REPORTER Item Tree, the Update Capture button will always update to switch to that Item type (Image, Movie, or GIF). Left-clicking the Update Capture button will then replace the current capture with one of the same type without the need to use the drop-down menu again. The drop-down menu can be used if switching type (e.g. PNG Image to MP4 Movie) is desired.

Settings such as frame rate and quality are determined by their current status in the Images -> Write -> Movies panel so be sure to check these before performing a Capture.

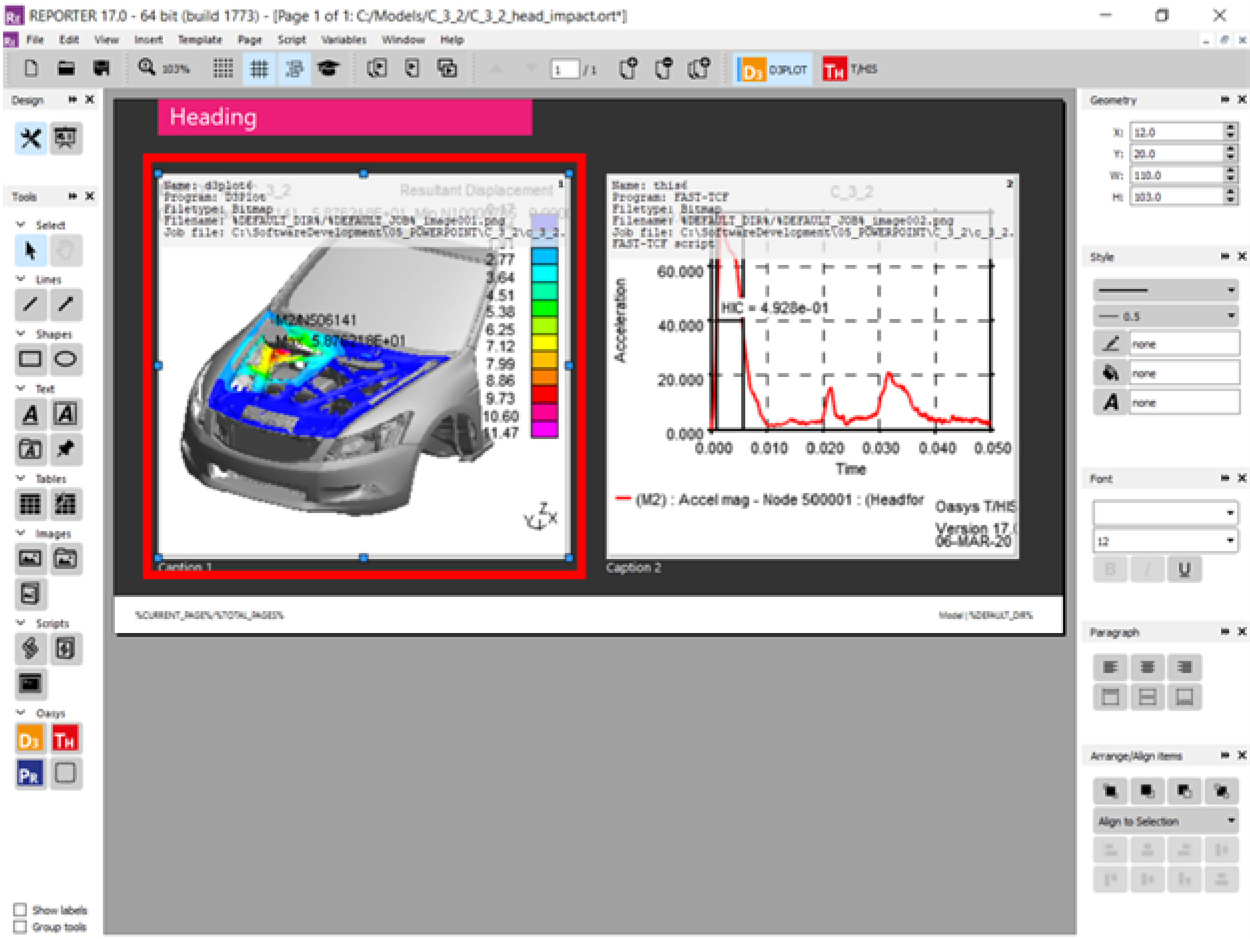


19.4. Reload

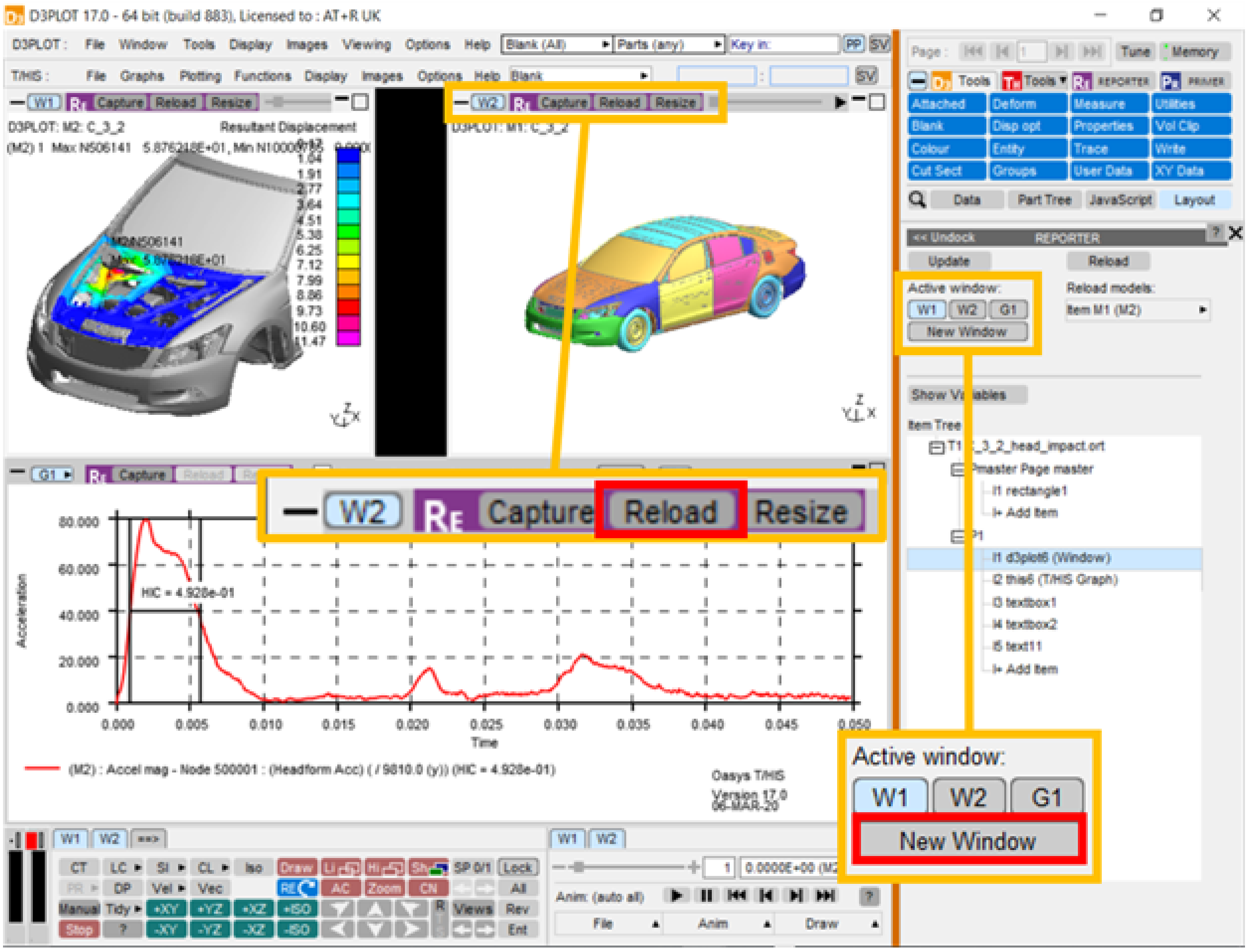
Reload

Existing REPORTER items can be reloaded back into D3PLOT or T/HIS. Items captured from graphs in the D3PLOT->T/HIS link are treated the same as items captured from standalone T/HIS. As such, they can each be reloaded either into D3PLOT or T/HIS.

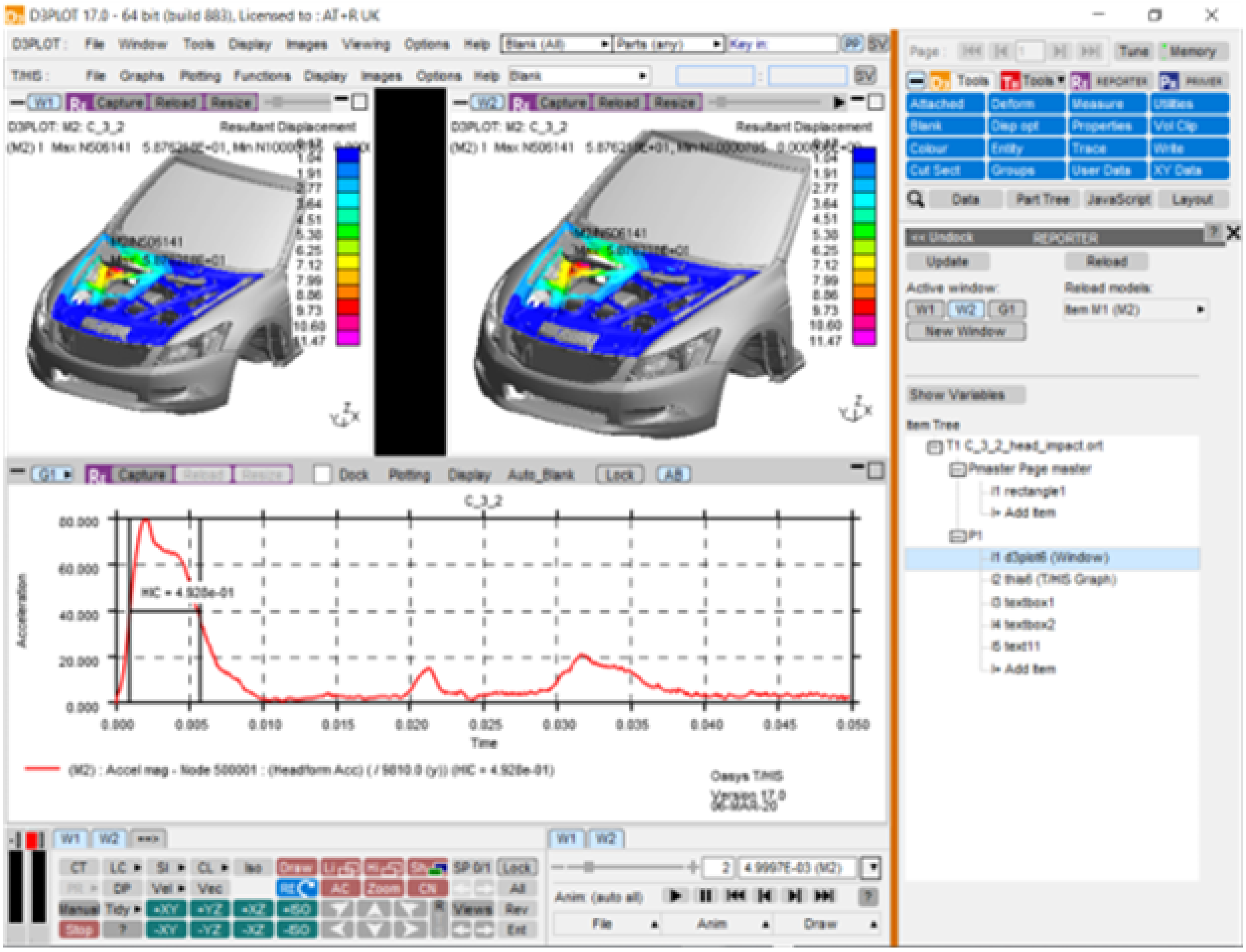
First select the item in REPORTER that you want to reload.



Then either press reload at the top of the target window, or select 'New Window' in the Active window list.



This will clear the target window, open the relevant models, not opening them again if they are already open in the session, then load the stored item information, reproducing the capture.

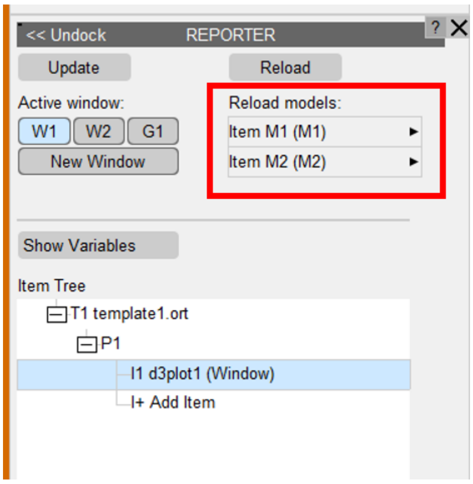


19.4.1. Reload Models

Reload Models

The models used in an existing item are listed in the Reload models list. The models will be listed as Item Mn, where n is the index of the model in the item, not of the model in the session. If the model is also open in the current session, then the model ID in the current session will be displayed in brackets.

Each entry in the list has a popup attached, allowing the model to replaced either by a model in the current session or by browsing for a model. This will not change the models stored in the item, but instead when the item is reloaded into the current session the replacement models will be used. The resulting window will then need to be captured, either into a new item or to overwrite the original.



19.5. Generate

Generate

Once a complete template has been created, it can be generated using File >> Generate in REPORTER. This will generate in an existing session if there is one, otherwise a new session will be started. T/HIS items will be generated in standalone T/HIS, unless the T/HIS link is already open in D3PLOT, in which case they will generate in the link. It is faster to generate in standalone T/HIS.

19.6. Variables

Variables

Variables can be added to both D3PLOT and T/HIS items, allowing data related to the capture to be made available in REPORTER. The REPORTER panel can be undocked and expanded to display the variables list by selecting Show Variables.

For T/HIS items, variables can be added containing properties of any of the curves in the selected graph or all the curves combined using the All Curves option. By default, T/HIS items will have variables for the MAX and MIN values taken over all curves in the selected graph. When selecting the curve for a newly created variable using the curve popup, curves are referred to as ICn, meaning Item Curve n, where n is the index of the curve in the selected graph. The curve label and number in the current session are also displayed in the popup.

For D3PLOT items, variables can be added for the MAX and MIN values of any of the plotted data components on any of the models. By default, D3PLOT items will have variables for the MAX and MIN values of all plotted data components for each model in the selected window.

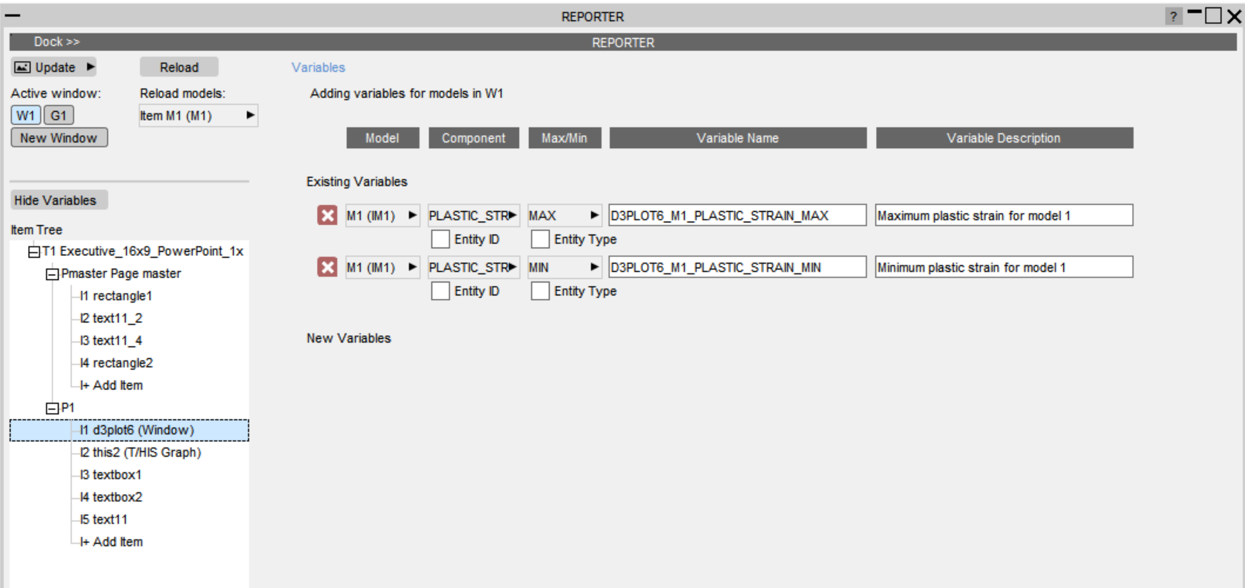
Variables can be added using the + button and deleted using the X button next to the row.

Initially, variables will appear under New Variables until the item is captured, when they will move to Existing Variables. Variables will be given default names based on their item name in REPORTER (e.g. d3plot6_1, this4), variable type and model/curve that they relate to. However, these names and descriptions can be manually edited.

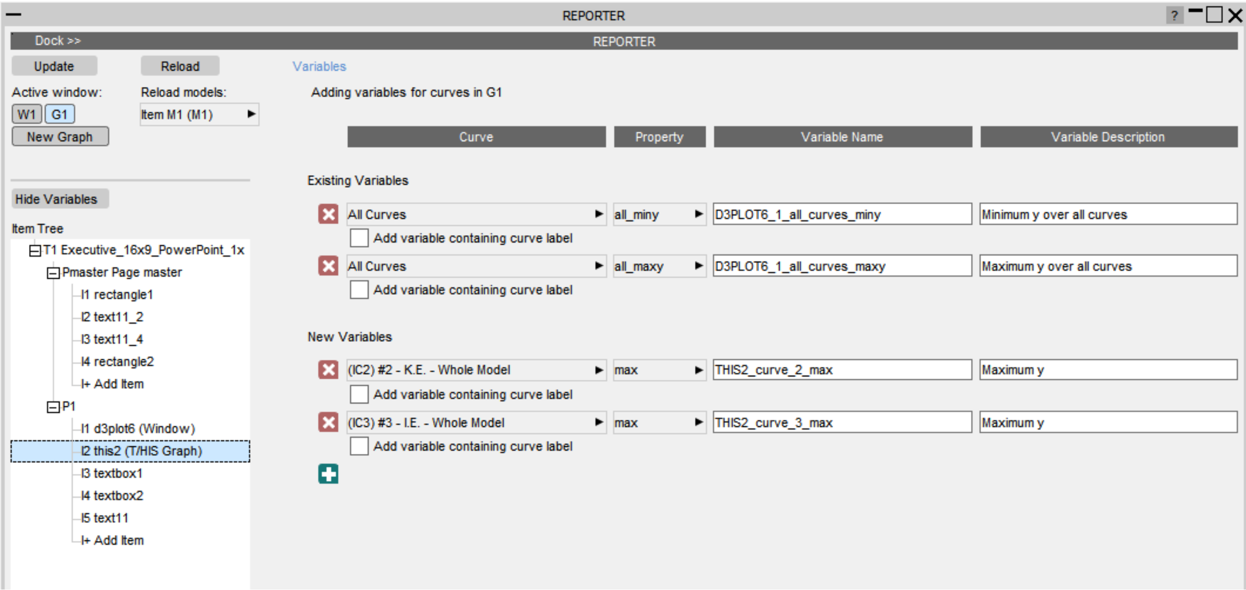
For D3PLOT items, the Entity ID and Entity Type tickboxes can be used to create additional variables to contain this information. These will have the same name as the original variable with either _ENT_ID or _ENT_TYPE appended.

For T/HIS items, the Add variables containing curve label tickbox will create an additional variable containing the curve label of the relevant curve, with _LABEL appended to the name.

Example of a D3PLOT item with two existing variables, referring to models in Window 1.



Example of a T/HIS item with two new variables and two existing variables, referring to curves in Graph 1.



19.7. Exceptions to the Oasys Suite 17.0 Method and Existing Templates from Oasys Suite 16.0 and Earlier

Exceptions to the Oasys Suite 17.0 Method and Existing Templates from Oasys Suite 16.0 and Earlier

There are some item types that are not supported in the new Oasys Suite 17.0 method. In this case, the Oasys Suite 16.0 method will be used and nothing will have changed. These are:

- T/HIS JavaScript items
- Items containing multiple graphs/windows

Any item can be captured and generated using the Oasys Suite 16.0 method by selecting the Capture and generate this item using the old method option in the object information in REPORTER.

Existing Oasys Suite 16.0 and earlier templates should work exactly as they used to. All items will use the Oasys Suite 17.0 method unless they meet one of the specified exceptions above. This gives some additional benefits:

- When generating the report, all supported items will be generated in the same session, without opening the same models multiple times. This will make the process faster.
- The report can be edited interactively using all the perks of the Oasys Suite 17.0 method.

20. Appendices

20.1. A. Programme Limitations

A. PROGRAMME LIMITATIONS

D3PLOT has certain inherent limitations because of its internal structure. These are:

20.1.1. Maximum Number of Nodes, Elements and Materials

Maximum number of nodes, elements and materials

Maximum number of nodes is	approx 670e6	These values are dictated by the theoretical limits upon internal storage. Contemporary machines will not have nearly enough memory to reach these limits.
Maximum number of elements is	approx 350e6	
Maximum number of parts is approx	50e6	

These are the maximum ***quantities*** that may be processed.

The maximum external ***label*** ids are limited only by output field width, so that the permitted external label range is 1 - 2**63 (approx 9e18).

In practice "small" format LS-DYNA is limited to 99,999,999, and "large" format LS-DYNA to 999,999,999,999,999.

20.1.2. Maximum Results Dataset Size that can be Read

Maximum results dataset size that can be read

Maximum individual file size:	9 Exawords	That is approximately 3.6e19 bytes in single precision (4 bytes/word) or 7.2e19 bytes in double precision (8 bytes/word).
Maximum overall dataset size:	9 Exawords	

20.1.3. Maximum Problem Size

Maximum problem size

Memory management in D3PLOT is "dynamic". This means that space is requested from the operating system as it is required and, theoretically, problems up to size limits above can be processed. The only realistic limit will be the ability of your computer to provide memory and processing power.

In practice a typical small workstation will easily process models of 500,000 elements, and addition of more memory and page file (swap) space will raise this limit to several million elements. For example processing a model with 170,000 shells on the author's machine required a total of about 55MBytes of memory. However large animations consume memory, and this may impose a limit upon effective problem size.

Most 32 bit operating systems impose a limit of 2GBytes on any process size, and this will tend to impose the upperbound on these systems. This is due to the maximum memory addresses can be stored in a 32 bit integer (4GB), and the needs of the operating system itself. An exception is x86 hardware under Windows and Linux which, with suitable configuration, will allow 3GB of memory to be used.

On 64 bit systems maximum memory usage is effectively unlimited, since a 64 bit integers permit up to 1.8e19 bytes (16,000,000 Terabytes) to be addressed. The maximum process size will be dictated by the amount of physical and virtual memory available.

20.1.4. Maximum Animation Sequence

Maximum animation sequence

Up to 2^{**31} (roughly 2e9) frames may be generated, saved and displayed - subject to memory limitations.

The amount of information, the product of <amount of graphics information> x <number of frames>, is limited only by the amount of memory available.

20.2. B. oa_pref File: Setting User Preferences

B. OA_PREF FILE: SETTING USER PREFERENCES

It is possible to set "user preferences" in the **`oa_pref`** file outside the programme. This file allows customisation of many aspects of the programme: for example colours, function key operations, display attributes, laser file settings, and so on. This can be done in three ways:

1. From Options > Edit Prefs in D3PLOT
2. From the Preferences editor called from the Shell
3. By hand-editing the preference file.

"oa_pref" naming convention and locations

The file is called "oa_pref". It is looked for in the following places in the order given:

- The optional administration directory defined by the environmental variable (**`$OA_ADMIN`** or **`$OA_ADMIN_xx`** where xx is the release number).
- The site-wide installation directory defined by the environment variable (**`$OA_INSTALL`**)
- The user's home directory: **`$HOME`** (Unix/Linux) or **`%USERPROFILE%`** (Windows)
- The current working directory

See Installation organisation for an explanation of the directory structure.

All four files are read (if they exist) and the last preference read will be the one used, so the file can be customised for a particular job or user at will.
Files do not have to exist in any of these locations, and if none exists the programme defaults will be used.

On Unix and Linux

> **`$HOME`** on Unix and Linux is usually the home directory specified for each user in the system password file.
> The shell command " **`printenv`** " (or on some systems " **`setenv`** ") will show the value of this variable if set.
> If not set then it is defined as the " **`~`** " directory for the user. The command " **`cd; pwd`** " will show this.

On Windows

> **`%USERPROFILE%`** on Windows is usually **`C:\Documents and Settings\<user id> \`**
> Issuing the " **`set`** " command from an MS-DOS prompt will show the value of this and other variables.

Generally speaking you should put

- Organisation-wide options in the version in $OA_ADMIN_xx and/or $OA_INSTALL,
- User-specific options in $HOME / %USERPROFILE%
- Project-specific options in the current working directory.

The file contains preferences for the SHELL (lines commencing shell*), THIS (lines commencing this*), D3PLOT (lines commencing d3plot*), PRIMER (lines commencing primer*)and REPORTER (lines commencing reporter*). All lines take the format <preference name> <preference value>.

The general copy of the preference file should be present in the $OA_ADMIN_xx and/or $OA_INSTALL directory. This should contain the preferences most suitable for all software users on the system.

An individual's specific preferences file can be stored in the individual's home area. This can be used to personally customise the software to the individual's needs.

Whenever one of the programs whose preferences can be stored in the oa_pref file is fired up, the program will take preferences first from the general preference file in the $OA_ADMIN_xx directory (if it exists) then the $OA_INSTALL directory, then from the file in the user's home area, then from the current working directory.

Preferences defined in the general oa_pref file can be modified in the user's personal file but they can't be removed by it.

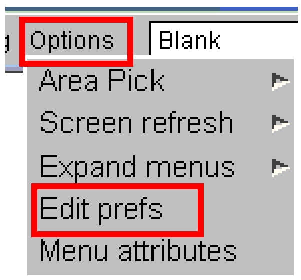
From Oasys Suite 9.4 onwards preferences can be locked. If a preference is locked it cannot be changed in an oa_pref file in a more junior directory. To lock a preference use the syntax **'d3plot#'** rather than **'d3plot*'** .

20.2.1. The Interactive Preferences Editor

The interactive Preferences Editor

You are free to edit oa_pref files by hand, but there is an interactive "Preferences Editor" that may be called from within D3PLOT that makes the job much easier.

It is started by **Options, Edit Prefs:**



The preferences editor reads an XML file that contains all possible preferences and their valid options, and allows you to change them at will. In this example the user is changing the background colour in D3PLOT.

Note that changes made in the Preferences editor will not affect the current session of D3PLOT, they will only take effect the next time it is run.

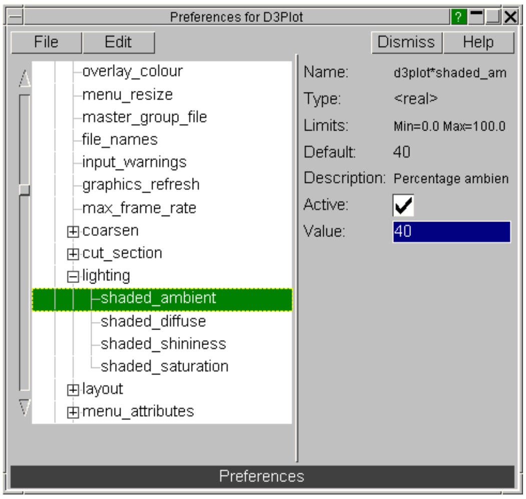
If you have write permission on the oa_pref file in the $OASYS directory you will be asked if you want to update that file, otherwise you will only be given the option of updating your own file in your $HOME / $USERPROFILE directory.

In this example the user is changing the ambient light intensity.

The option is "active" (ie present in the oa_pref file) and currently is set to 40.

Usage is:

- Select an option in the Tree on the left hand side
- Make it active / inactive
- If active select a value from the popup, or type in a value if necessary



The colour of the highlighting in the left hand side tree is significant:

Green	Means that the option has been read from your $HOME/$USERPROFILE file.
Red	Means that the option has been read from the $OA_INSTALL file.
Magenta	Means that the option had been read from the $OA_ADMIN file.

In either event, regardless of the data source, the updated option will be written to the file chosen when you started the preferences editor.

Because of the order of file reading (see above), and option read from the master $OASYS file, amended, and written to your local $HOME file will take precedence when you next run D3PLOT.

20.2.2. Locking Preferences

Locking Preferences

From Oasys Suite 9.4 onwards preferences can be locked. Beside each option in the preference editor is a padlock symbol. If the symbol is green then the option is unlocked, if it is red then it is locked. If a preference option has been locked in a file that the user can not modify then an error message will be generated if the user tries to edit that option.

If a user manually edits the "oa_pref" file to try and set an option that has been locked in another preference file then the option will be ignored in the users preference file.

20.2.3. Format of the oa_pref File

Format of the oa_pref file

Entries are formatted in the following way: **<programme>*<option>: <setting>**
For example: **d3plot*laser_paper_size: A4**

The rules for formatting are:

- The **<programme>*<option>:** string must start at column 1;
- This string must be in lower case, and must not have any spaces in it.
- The **<setting>** must be separated from the string by at least one space.
- Lines starting with a " **#** " are treated as comments and are ignored.

(Users accustomed to setting the attributes of their window manager with the **.Xdefaults** file will recognise this format and syntax.)

"oa_pref" arguments valid for D3PLOT.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
write_checkpoint_files	<logical>	Record checkpoint files for the D3PLOT session.	TRUE, FALSE	FALSE
checkpoint_dir	<string>	Directory for checkpoint files, or "none" to suppress them altogether		<none>
show_checkpoint_files	<logical>	Show checkpoint playback panel upon D3PLOT startup.	TRUE, FALSE	FALSE
error_handler	<string>	how to handle errors and exceptions	no_action, mini_dump, trap_continue, trace_exit	mini_dump
fix_cwd	<string>	Option to fix the CWD (DEFAULT/START_IN/)		<none>
start_in	<string>	Directory to start D3PLOT in		<none>
splash_screen_seen	<real>	Most recent version (as major.minor, eg 17.1) for which a splash screen has been seen		0.0
show_license_warning	<logical>	Display Window containing License System messages	TRUE, FALSE	TRUE
save_window_positions	<logical>	Save position of undocked windows between sessions	TRUE, FALSE	TRUE

annotations				
annotation_desc_size	<string>	Font size for the annotation playback tab	8, 10, 12, 14, 18, 24, Default	Default

The following options permit the default automatic blanking behaviour of contact segments and spotwelds to be modified. The default behaviour is to automatically blank contact segments and spotwelds if the associated element is blanked. When a contact segment is coincident with both a Shell (2D) and a Solid/Thick Shell (3D) then the default is to blank the segment based on the 2D shell.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
auto_blank_segments	<string>	Automatically blank contact segments when underlying elements are blanked	ON, OFF	ON
blank_segments_2d_3d	<string>	Blank contact segments with Shells (2D) or Solids/Thick Shells (3D)	2D, 3D	2D
auto_blank_spotwelds	<string>	Automatically blank spotwelds when underlying elements are blanked	ON, OFF	ON

The following options can be used to control what D3PLOT does if it it encounters nodal coordinates that are very large compared to a models undeformed bounding box. This can be useful if you have a model where nodes have gone flying off due to element deletion or other problems. By default D3PLOT will check the nodal coordinates for each state and if a node has moved to a location more than 1000 times the size of the models initial bounding box then the nodes coordinates will automatically be adjusted. This check can be turned off and the default factor of 1000 can also be changed. See **Clamp Data** for more information.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
clamp_nodes	<logical>	Controls if D3PLOT automatically clamps the coordinates of nodes that have moved .	TRUE, FALSE	TRUE
clamp_node_factor	<integer>	Model bounding box factor for clamping nodes	1 - 2147483646	1000
disps_64_bit	<logical>	If TRUE and the source file is 64 bit D3PLOT calculates displacements at full 64 bit precision	TRUE, FALSE	TRUE

The following options define the colours used in plots, thereby affecting plot appearance.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
background_mode	<string>	Draw background using SOLID or FADED colours	SOLID, FADED	SOLID
background_colour	<string>	Background colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	BLACK

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY, AUTO	
background_bottom_colour	<string>	Background bottom colour for faded backgrounds (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY, AUTO	AUTO
overlay_colour	<string>	Overlay colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA,	GREY

			GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY, ELEMENT	
overlay_transparency	<string>	Use Transparency for Overlay	ON, OFF	ON
text_colour	<string>	Text colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	WHITE
user_colours_file	<string>	Location of the user-defined colours XML file.		<none>
save_colours_on_exit	<logical>	Automatically save the user colours XML file when the program exits.	TRUE, FALSE	TRUE

The following options permit the default cut-section method and space system to be configured. Users who wish to default to compatibility with the LS-DYNA section definition method may wish to use these options. See **CUT_SECTIONS** for more information.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
cut_section_cap2d	<string>	Method of shell element cut section capping	NO_CAPPING, TRUE_THICKNESS, FIXED_THICKNESS, PART_CONTACT_OPTT	FIXED_THICKNESS
cut_section_cap2d_fac	<real>	True-thickness factor for shell element cut section capping		1
cut_section_cap2d_val	<real>	Fixed thickness value for shell element cut section capping		10
cut_section_display	<string>	Cut section plane display	OFF, WIREFRAME, TRANSPARENT	OFF
cut_section_fsys	<string>	Cut section force and moment computation system	AUTOMATIC, GLOBAL, LOCAL	AUTOMATIC
cut_section_per_state	<string>	Draw true-thickness shell element capping for each state (rather than using the initial value for all states)	FALSE, TRUE	TRUE
cut_section_space	<string>	Default cut section space system	BASIC, DEFORMED, SCREEN	DEFORMED
cut_section_type	<string>	Default cut section definition method	OV_ORIGIN_AND_VECTORS, N3_THREE_NODES, X_CONSTANT, Y_CONSTANT,	X_CONSTANT

			Z_CONSTANT, LS_DYNA_METHOD	
cut_section_zero_orig	<string>	Use the global origin [0, 0, 0] as default cut origin instead of model centre	FALSE, TRUE	FALSE
csd_update_bands	<string>	Update auto contour bands during cut section drag	OFF, ON	OFF
csd_update_maxmin	<string>	Update max and min display during cut section drag	OFF, ON	OFF
csd_contour_face	<string>	Draw contours on cut face during cut section drag	OFF, 3D_ONLY, 2D_ONLY, ON	OFF
csd_display_face	<string>	Draw interpolated cut face during cut section drag	OFF, 3D_ONLY, 2D_ONLY, ON	OFF
cut_section_capping_colour	<string>	Cut section capping colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	DEFAULT

			GREY, LIGHT_GREY, DEFAULT	
cut_section_outline_switch	<string>	Cut section outline switch	OFF, ON	OFF
cut_section_outline_width_val	<integer>	Cut section outline width	1 - 10	1
cut_section_outline_colour	<string>	Cut section outline colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW

The following options control the default location and name of where D3PLOT looks for model database files.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
database_dir	<string>	Directory to look in for model database (XML) files		<none>
database_file	<string>	Default model database (XML) file		<none>
database_expand	<integer>	Number of levels to automatically expand in model database tree (-1 ALL)	-1 - 2147483646	0

The following options affect the appearance and behaviour of data-bearing (usually contour) plots (see DATA COMPONENTS-BASIC).

Preference	**Type**	**Description**	**Valid arguments**	**Default**
default_component	<string>	Valid data component name		PLASTIC_STRAIN
clamp_data	<logical>	Controls whether D3PLOT clamps data values to be within a defined magnitude.	TRUE, FALSE	TRUE
clamp_max_value	<real>	Data clamping magnitude		1e18

clamp_to_zero	<logical>	Whether clamped data (|value| > max_magnitude) is reset to zero.	TRUE, FALSE	TRUE
contour_average_switch	<string>	Average contours at nodes	ON, OFF	ON
contour_part_ignored_switch	<string>	Average regardless of part number	ON, OFF	ON
contour_blanking_ignored_switch	<string>	Average regardless of element blanking	ON, OFF	ON
contour_clipping_ignored_switch	<string>	Average regardless of element clipping	ON, OFF	ON
contour_levels	<integer>	Number of contour levels	2 - 16	13
contour_bar_3d_faces	<string>	Faces of 3D Elements used for contour bar range	VISIBLE, ALL	VISIBLE
contour_range_label	<string>	Contour range label switch	ON, OFF	ON
contour_colours				
user_contour_colour_levels_1_band_1	<string>	Colour for contour band 1 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_2_band_1	<string>	Colour for contour band 1 when	WHITE, BLACK, RED,	GREEN

		number of levels is 2 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_2_band_2	<string>	Colour for contour band 2 when number of levels is 2 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	RED

			GREY, LIGHT_GREY	
user_contour_colour_levels_3_band_1	<string>	Colour for contour band 1 when number of levels is 3 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
user_contour_colour_levels_3_band_2	<string>	Colour for contour band 2 when number of levels is 3 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	GREEN

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_3_band_3	<string>	Colour for contour band 3 when number of levels is 3 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_4_band_1	<string>	Colour for contour band 1 when number of levels is 4 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	BLUE

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_4_band_2	<string>	Colour for contour band 2 when number of levels is 4 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_4_band_3	<string>	Colour for contour band 3 when number of levels is 4 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	YELLOW

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_4_band_4	<string>	Colour for contour band 4 when number of levels is 4 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_5_band_1	<string>	Colour for contour band 1 when number of levels is 5 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	BLUE

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_5_band_2	<string>	Colour for contour band 2 when number of levels is 5 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_5_band_3	<string>	Colour for contour band 3 when number of levels is 5 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	YELLOW

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_5_band_4	<string>	Colour for contour band 4 when number of levels is 5 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_5_band_5	<string>	Colour for contour band 5 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	MAGENTA

		5 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_6_band_1	<string>	Colour for contour band 1 when number of levels is 6 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	BLUE

			GREY, LIGHT_GREY	
user_contour_colour_levels_6_band_2	<string>	Colour for contour band 2 when number of levels is 6 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_6_band_3	<string>	Colour for contour band 3 when number of levels is 6 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	GREEN

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_6_band_4	<string>	Colour for contour band 4 when number of levels is 6 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
user_contour_colour_levels_6_band_5	<string>	Colour for contour band 5 when number of levels is 6 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	RED

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_6_band_6	<string>	Colour for contour band 6 when number of levels is 6 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
user_contour_colour_levels_7_band_1	<string>	Colour for contour band 1 when number of levels is 7 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	BLUE

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_7_band_2	<string>	Colour for contour band 2 when number of levels is 7 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_7_band_3	<string>	Colour for contour band 3 when number of levels is 7 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	GREEN/CYAN

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_7_band_4	<string>	Colour for contour band 4 when number of levels is 7 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_7_band_5	<string>	Colour for contour band 5 when number of levels is 7 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	YELLOW

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_7_band_6	<string>	Colour for contour band 6 when number of levels is 7 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_7_band_7	<string>	Colour for contour band 7 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	MAGENTA

		7 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_8_band_1	<string>	Colour for contour band 1 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	BLUE

			GREY, LIGHT_GREY	
user_contour_colour_levels_8_band_2	<string>	Colour for contour band 2 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_8_band_3	<string>	Colour for contour band 3 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	GREEN/CYAN

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_8_band_4	<string>	Colour for contour band 4 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_8_band_5	<string>	Colour for contour band 5 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	YELLOW

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_8_band_6	<string>	Colour for contour band 6 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
user_contour_colour_levels_8_band_7	<string>	Colour for contour band 7 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	RED

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_8_band_8	<string>	Colour for contour band 8 when number of levels is 8 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
user_contour_colour_levels_9_band_1	<string>	Colour for contour band 1 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	BLUE

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_9_band_2	<string>	Colour for contour band 2 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_9_band_3	<string>	Colour for contour band 3 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	GREEN/CYAN

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_9_band_4	<string>	Colour for contour band 4 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_9_band_5	<string>	Colour for contour band 5 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	YELLOW/GREEN

		9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_9_band_6	<string>	Colour for contour band 6 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	YELLOW

			GREY, LIGHT_GREY	
user_contour_colour_levels_9_band_7	<string>	Colour for contour band 7 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
user_contour_colour_levels_9_band_8	<string>	Colour for contour band 8 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	RED

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_9_band_9	<string>	Colour for contour band 9 when number of levels is 9 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
user_contour_colour_levels_10_band_1	<string>	Colour for contour band 1 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	BLUE

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_10_band_2	<string>	Colour for contour band 2 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_10_band_3	<string>	Colour for contour band 3 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	GREEN/CYAN

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_10_band_4	<string>	Colour for contour band 4 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_10_band_5	<string>	Colour for contour band 5 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	YELLOW/GREEN

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_10_band_6	<string>	Colour for contour band 6 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
user_contour_colour_levels_10_band_7	<string>	Colour for contour band 7 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	RED/ORANGE

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_10_band_8	<string>	Colour for contour band 8 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_10_band_9	<string>	Colour for contour band 9 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	RED/MAGENTA

		1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_10_band_10	<string>	Colour for contour band 10 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)0	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	MAGENTA

			GREY, LIGHT_GREY	
user_contour_colour_levels_11_band_1	<string>	Colour for contour band 1 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
user_contour_colour_levels_11_band_2	<string>	Colour for contour band 2 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	CYAN/BLUE

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_11_band_3	<string>	Colour for contour band 3 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_11_band_4	<string>	Colour for contour band 4 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	GREEN/CYAN

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_11_band_5	<string>	Colour for contour band 5 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_11_band_6	<string>	Colour for contour band 6 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	YELLOW/GREEN

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_11_band_7	<string>	Colour for contour band 7 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
user_contour_colour_levels_11_band_8	<string>	Colour for contour band 8 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	RED/ORANGE

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_11_band_9	<string>	Colour for contour band 9 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_11_band_10	<string>	Colour for contour band 10 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	RED/MAGENTA

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_11_band_11	<string>	Colour for contour band 11 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)1	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
user_contour_colour_levels_12_band_1	<string>	Colour for contour band 1 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	BLUE

		1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_12_band_2	<string>	Colour for contour band 2 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	CYAN/BLUE

			GREY, LIGHT_GREY	
user_contour_colour_levels_12_band_3	<string>	Colour for contour band 3 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_12_band_4	<string>	Colour for contour band 4 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	GREEN/CYAN

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_12_band_5	<string>	Colour for contour band 5 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_12_band_6	<string>	Colour for contour band 6 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	YELLOW/GREEN

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_12_band_7	<string>	Colour for contour band 7 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
user_contour_colour_levels_12_band_8	<string>	Colour for contour band 8 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	RED/ORANGE

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_12_band_9	<string>	Colour for contour band 9 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	DARK_ORANGE
user_contour_colour_levels_12_band_10	<string>	Colour for contour band 10 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	RED

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_12_band_11	<string>	Colour for contour band 11 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/MAGENTA
user_contour_colour_levels_12_band_12	<string>	Colour for contour band 12 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)2	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	MAGENTA

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_13_band_1	<string>	Colour for contour band 1 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
user_contour_colour_levels_13_band_2	<string>	Colour for contour band 2 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	CYAN/BLUE

		1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_13_band_3	<string>	Colour for contour band 3 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	LIGHT_BLUE

			GREY, LIGHT_GREY	
user_contour_colour_levels_13_band_4	<string>	Colour for contour band 4 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_13_band_5	<string>	Colour for contour band 5 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	GREEN/CYAN

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_13_band_6	<string>	Colour for contour band 6 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_13_band_7	<string>	Colour for contour band 7 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	YELLOW/GREEN

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_13_band_8	<string>	Colour for contour band 8 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
user_contour_colour_levels_13_band_9	<string>	Colour for contour band 9 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	RED/ORANGE

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_13_band_10	<string>	Colour for contour band 10 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	DARK_ORANGE
user_contour_colour_levels_13_band_11	<string>	Colour for contour band 11 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	RED

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_13_band_12	<string>	Colour for contour band 12 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/MAGENTA
user_contour_colour_levels_13_band_13	<string>	Colour for contour band 13 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)3	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	MAGENTA

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_1	<string>	Colour for contour band 1 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
user_contour_colour_levels_14_band_2	<string>	Colour for contour band 2 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	CYAN/BLUE

		1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_3	<string>	Colour for contour band 3 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	LIGHT_BLUE

			GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_4	<string>	Colour for contour band 4 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
user_contour_colour_levels_14_band_5	<string>	Colour for contour band 5 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	GREEN/CYAN

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_6	<string>	Colour for contour band 6 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
user_contour_colour_levels_14_band_7	<string>	Colour for contour band 7 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	YELLOW/GREEN

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_8	<string>	Colour for contour band 8 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
user_contour_colour_levels_14_band_9	<string>	Colour for contour band 9 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	RED/ORANGE

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_10	<string>	Colour for contour band 10 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	DARK_ORANGE
user_contour_colour_levels_14_band_11	<string>	Colour for contour band 11 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	RED

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_12	<string>	Colour for contour band 12 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/MAGENTA
user_contour_colour_levels_14_band_13	<string>	Colour for contour band 13 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	MAGENTA

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_14_band_14	<string>	Colour for contour band 14 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)4	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
user_contour_colour_levels_15_band_1	<string>	Colour for contour band 1 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	BLUE

		1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_2	<string>	Colour for contour band 2 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	CYAN/BLUE

			GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_3	<string>	Colour for contour band 3 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	LIGHT_BLUE
user_contour_colour_levels_15_band_4	<string>	Colour for contour band 4 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	CYAN

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_5	<string>	Colour for contour band 5 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN/CYAN
user_contour_colour_levels_15_band_6	<string>	Colour for contour band 6 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	GREEN

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_7	<string>	Colour for contour band 7 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW/GREEN
user_contour_colour_levels_15_band_8	<string>	Colour for contour band 8 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	YELLOW

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_9	<string>	Colour for contour band 9 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
user_contour_colour_levels_15_band_10	<string>	Colour for contour band 10 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	DARK_ORANGE

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_11	<string>	Colour for contour band 11 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_15_band_12	<string>	Colour for contour band 12 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	RED/MAGENTA

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_13	<string>	Colour for contour band 13 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
user_contour_colour_levels_15_band_14	<string>	Colour for contour band 14 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	MAGENTA

		1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_15_band_15	<string>	Colour for contour band 15 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)5	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	MAGENTA

			GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_1	<string>	Colour for contour band 1 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
user_contour_colour_levels_16_band_2	<string>	Colour for contour band 2 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	CYAN/BLUE

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_3	<string>	Colour for contour band 3 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	LIGHT_BLUE
user_contour_colour_levels_16_band_4	<string>	Colour for contour band 4 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREE	CYAN

			N, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_5	<string>	Colour for contour band 5 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN/CYAN
user_contour_colour_levels_16_band_6	<string>	Colour for contour band 6 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW,	GREEN

			GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_7	<string>	Colour for contour band 7 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW/GREEN
user_contour_colour_levels_16_band_8	<string>	Colour for contour band 8 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN,	YELLOW

			LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_9	<string>	Colour for contour band 9 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
user_contour_colour_levels_16_band_10	<string>	Colour for contour band 10 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA	DARK_ORANGE

			, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_11	<string>	Colour for contour band 11 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
user_contour_colour_levels_16_band_12	<string>	Colour for contour band 12 when number of levels is	WHITE, BLACK, RED, GREEN, BLUE,	RED/MAGENTA

		1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_13	<string>	Colour for contour band 13 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY,	MAGENTA

			GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_14	<string>	Colour for contour band 14 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
user_contour_colour_levels_16_band_15	<string>	Colour for contour band 15 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN,	MAGENTA

			MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
user_contour_colour_levels_16_band_16	<string>	Colour for contour band 16 when number of levels is 1 (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
old_contour_colours				
contour_colour_levels_1_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE,	GREEN

			DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_2_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
contour_colour_levels_2_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE,	RED

			DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_3_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_3_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN,	GREEN

			INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_3_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
contour_colour_levels_4_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN,	BLUE

			MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_4_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN

contour_colour_levels_4_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
contour_colour_levels_4_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	RED

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_5_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_5_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	GREEN

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_5_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
contour_colour_levels_5_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	RED

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_5_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_6_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	BLUE

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_6_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
contour_colour_levels_6_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	GREEN

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_6_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
contour_colour_levels_6_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	RED

D3PLOT

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_6_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA

contour_colour_levels_7_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_7_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	CYAN

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_7_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN/CYAN
contour_colour_levels_7_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	GREEN

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_7_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
contour_colour_levels_7_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	RED

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_7_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_8_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	BLUE

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_8_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
contour_colour_levels_8_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	GREEN/CYAN

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_8_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
contour_colour_levels_8_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	YELLOW

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_8_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE

contour_colour_levels_8_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
contour_colour_levels_8_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	MAGENTA

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_9_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_9_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	CYAN

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_9_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN/CYAN
contour_colour_levels_9_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	GREEN

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_9_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW/GREEN
contour_colour_levels_9_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	YELLOW

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_9_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
contour_colour_levels_9_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	RED

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_9_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_10_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	BLUE

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_10_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN

contour_colour_levels_10_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN/CYAN
contour_colour_levels_10_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	GREEN

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_10_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW/GREEN
contour_colour_levels_10_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	YELLOW

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_10_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
contour_colour_levels_10_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	RED

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_10_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/MAGENTA
contour_colour_levels_10_band_10	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	MAGENTA

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_11_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_11_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	CYAN/BLUE

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_11_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
contour_colour_levels_11_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	GREEN/CYAN

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_11_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN

contour_colour_levels_11_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW/GREEN
contour_colour_levels_11_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	YELLOW

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_11_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
contour_colour_levels_11_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	RED

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_11_band_10	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/MAGENTA
contour_colour_levels_11_band_11	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	MAGENTA

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_12_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_12_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	CYAN/BLUE

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_12_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
contour_colour_levels_12_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	GREEN/CYAN

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_12_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
contour_colour_levels_12_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	YELLOW/GREEN

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_12_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW

contour_colour_levels_12_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
contour_colour_levels_12_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	DARK_ORANGE

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_12_band_10	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
contour_colour_levels_12_band_11	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	RED/MAGENTA

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_12_band_12	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_13_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	BLUE

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_13_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN/BLUE
contour_colour_levels_13_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	LIGHT_BLUE

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_13_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
contour_colour_levels_13_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	GREEN/CYAN

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_13_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
contour_colour_levels_13_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	YELLOW/GREEN

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_13_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW

contour_colour_levels_13_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
contour_colour_levels_13_band_10	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	DARK_ORANGE

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_13_band_11	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
contour_colour_levels_13_band_12	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	RED/MAGENTA

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_13_band_13	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_14_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	BLUE

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_14_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN/BLUE
contour_colour_levels_14_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	LIGHT_BLUE

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_14_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	CYAN
contour_colour_levels_14_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	GREEN/CYAN

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_14_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
contour_colour_levels_14_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	YELLOW/GREEN

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_14_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW

contour_colour_levels_14_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/ORANGE
contour_colour_levels_14_band_10	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	DARK_ORANGE

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_14_band_11	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
contour_colour_levels_14_band_12	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	RED/MAGENTA

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_14_band_13	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_14_band_14	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	MAGENTA

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_15_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_15_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	CYAN/BLUE

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_15_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	LIGHT_BLUE
contour_colour_levels_15_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	CYAN

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_15_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN/CYAN
contour_colour_levels_15_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	GREEN

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_15_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW/GREEN

contour_colour_levels_15_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
contour_colour_levels_15_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	RED/ORANGE

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_15_band_10	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	DARK_ORANGE
contour_colour_levels_15_band_11	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	RED

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_15_band_12	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/MAGENTA
contour_colour_levels_15_band_13	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	MAGENTA

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_15_band_14	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_15_band_15	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	MAGENTA

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_1	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
contour_colour_levels_16_band_2	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	CYAN/BLUE

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_3	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	LIGHT_BLUE
contour_colour_levels_16_band_4	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	CYAN

			YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_5	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN/CYAN

contour_colour_levels_16_band_6	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
contour_colour_levels_16_band_7	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN,	YELLOW/GREEN

			PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_8	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	YELLOW
contour_colour_levels_16_band_9	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK,	RED/ORANGE

			RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_10	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	DARK_ORANGE
contour_colour_levels_16_band_11	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGEN	RED

			TA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_12	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED/MAGENTA
contour_colour_levels_16_band_13	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANG	MAGENTA

			E, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_14	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_colour_levels_16_band_15	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO,	MAGENTA

			YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
contour_colour_levels_16_band_16	<string>	Deprecated: use user_contour_colour_... preference	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	MAGENTA
contour_ramp	<string>	Contour ramp colour palette	DEFAULT, VIRIDIS, PLASMA, MAGMA, INFERNO	DEFAULT

contour_number_format_type	<string>	Number format type for contour bar	AUTO, SCIENTIFIC, GENERAL, MANUAL	AUTO
contour_dec_places	<integer>	Number of decimal places to display on contour bar	1 - 9	3
contour_exponent	<integer>	Value of exponent to use on contour bar	-9 - 9	3
dyna_layer_order	<logical>	Layer selection is in LS-DYNA order	TRUE, FALSE	FALSE
show_maxmin	<string>	Max and Min data values drawn on plot	OFF, DATA, ALL	DATA
maxmin_number_format_type	<string>	Number format type for max min values	AUTO, SCIENTIFIC, GENERAL, MANUAL	AUTO
maxmin_dec_places	<integer>	Number of decimal places for max min values	1 - 9	3
maxmin_exponent	<integer>	Value of exponent to use for max min values	-9 - 9	3
nastran_strain_warning_message	<logical>	Display Nastran engineering shear strain warning message	TRUE, FALSE	TRUE
pemag_calculation_v12	<logical>	Whether the PEMAG calculation uses the v12 logic.	TRUE, FALSE	FALSE
signed_von_mises_format	<string>	Method for calculating sign of von Mises stress	PRINCIPAL, INVARIANT	PRINCIPAL
lode_param_tension_sign	<string>	Sign convention for lode parameter for generalised tension	POSITIVE, NEGATIVE, NOT_SET	NOT_SET
surface_output	<string>	Surface to output results	TOP, MIDDLE, BOTTOM, MAX, MIN, ALL	MIDDLE
frequency_surface_output	<string>	Default surface use to output results in FREQUENCY analysis	TOP, MIDDLE, BOTTOM, MAX, MIN, ALL	MIDDLE
struct_iso_resolution	<string>	Default resolution for structural ISO plot surfaces	8, 16	8
vol3_iso_resolution	<string>	Default resolution for Volume III ISO plot surfaces	8, 16	16

thick_shell_contour	<string>	Thick shell contouring options	INTERPOLATE, SIMPLE	INTERPOLATE
mixed_vector_data	<string>	Whether vector data drawn on non-'current' mode elements	SHOWN, NOT_SHOWN	NOT_SHOWN
vector_fill_colour	<string>	Colour used to fill 2d/3d elements in vector etc plot (hex code e.g. 0XA1B2C3 or core colour name e.g. RED)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREY
vector_fill_mode	<string>	Structure 'fill' mode in Vel, LC, etc plots	HIDDEN, SHADED	SHADED
limits_show_nc	<string>	Show non-contoured items when limiting values are on	OFF, ON	OFF
show_all_components	<logical>	Show all possible data components in XY Data and Write menus	TRUE, FALSE	FALSE
show_beam_component_warning	<logical>	Show warning when selecting integrated or resultant beam component	TRUE, FALSE	TRUE
ignore_beam_type	<string>	Plot integrated and resultant components on all	OFF, ON	OFF

		beam types when switch is on		

The following options affect the appearance and behaviour of single-point elements that have their data mapped onto a cellular mapping volume (see Mapping data onto a cellular grid for plotting).

Preference	**Type**	**Description**	**Valid arguments**	**Default**
cut_section_data				
mvol_abp_cut	<string>	Map cut-section data plots of Airbag Particle (ABP) data	OFF, ON	OFF
mvol_sph_cut	<string>	Map cut-section data plots of Smooth Particle Hydrodyanmics (SPH) data	OFF, ON	OFF
mvol_des_cut	<string>	Map cut-section data plots of Discrete Element Sphere (DES) data	OFF, ON	OFF
iso_surface_data				
mvol_abp_iso	<string>	Map ISO plots of Airbag Particle (ABP) data	OFF, ON	OFF
mvol_sph_iso	<string>	Map ISO plots of Smooth Particle Hydrodyanmics (SPH) data	OFF, ON	OFF
mvol_des_iso	<string>	Map ISO plots of Discrete Element Sphere (DES) data	OFF, ON	OFF
cell_size_options				
mvol_size_option	<string>	How mapping cell size is defined	CHAR_ELEMS, PERCENTAGE, FIXED, USER_DEF	CHAR_ELEMS
mvol_edge_els	<integer>	Characteristic #elements down edge	1 - 1000	5
mvol_edge_perc	<real>	Characteristic %age of model bounding box		5.0
mvol_edge_size	<real>	User-specified cell edge size		0.0
mvol_origin_x	<real>	X coordinate of user-defined grid origin		0.0
mvol_origin_y	<real>	Y coordinate of user-defined grid origin		0.0
mvol_origin_z	<real>	Z coordinate of user-defined grid origin		0.0
mvol_ncells_x	<integer>	Number of cells in X direction of user-defined grid		10
mvol_ncells_y	<integer>	Number of cells in Y direction of user-defined grid		10

mvol_ncells_z	<integer>	Number of cells in Z direction of user-defined grid		10
calculation_options				
mvol_calc_option	<string>	How element data is to be computed in a cell	SUM, AVERAGE, MAGNITUDE	SUM
mvol_dvol_option	<string>	Whether data in a cell is to be divided by cell volume	RAW, DIVIDE	RAW
smoothing_options				
mvol_smoothing	<string>	Whether data is to be smoothed across adjacent cells	OFF, ON	OFF
mvol_smooth_els	<integer>	Number of adjacent cells to smooth over	1 - 100	1
mvol_draw_underlying	<string>	Whether to draw the wireframe overlay of the elements themselves	OFF, ON	ON
mvol_draw_plane	<string>	Whether to contour data on the cut plane (if cutting active)	OFF, ON	ON
mvol_show_grid	<string>	Whether to show the mapping volume cellular grid	OFF, ON	OFF

Dynamic label settings.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
dynamic_label_format	<string>	Number format type for dynamic labels	AUTO, SCIENTIFIC, GENERAL	AUTO
dynamic_label_dec_places	<integer>	Number of decimal places to display on dynamic labels	0 - 9	3

The following options control the display of Element and Material triads. See CUT_SECTIONS for more information.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
triad_mode	<string>	Controls the display mode for Triad Symbols	TRIAD+LABELS, COLOURED_TRIAD, X-AXS_ONLY, Y-AXS_ONLY, Z-AXS_ONLY	TRIAD+LABELS
triad_size	<string>	Display triads using either a FIXED size or an AUTOMATICALY calculated size	FIXED, AUTOMATIC	FIXED
triad_fixed_size	<integer>	Size used for FIXED size triads	10 - 500	100

triad_line_width	<integer>	Line width used for drawing triads	1 - 10	2
triad_x_axis_colour	<string>	Coloured Triad X-axis colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	RED
triad_y_axis_colour	<string>	Coloured Triad Y-axis colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	GREEN
triad_z_axis_colour	<string>	Coloured Triad Z-axis colour (hex code e.g.	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA,	BLUE

		0XA1B2C3 or core colour name e.g. OLIVE)	YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	

The following set of options defines how various files are handled by D3Plot.
Option "master_group_file" defines a "master" group file to be read every time a new model is opened: (see GROUPS).
For information regarding the various file formats corresponding to the "read_<xxx>" options, see Reading Results.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
autocreate_ztf	<logical>	Create ZTF file automatically if required	TRUE, FALSE	FALSE
auto_open	<logical>	Controls if D3PLOT automatically opens a model as soon as it is selected with the file selector.	TRUE, FALSE	TRUE
blast_force_file_ext	<string>	Default file extension for Blast force file		*.blstfor
blast_force_file_name	<string>	Default name for Blast force file		blstfor
contact_force_file_ext	<string>	Default file extension for Contact force files		*.ctf

contact_force_file_name	<string>	Default name for Contact force file		ctfile
cpm_force_file_ext	<string>	Default file extension for CPM force files		*.cpm
cpm_force_file_name	<string>	Default name for CPM force file		cpmfor
delete_group_file	<logical>	Whether groups (.grp) file should be deleted on exit	TRUE, FALSE	FALSE
dem_force_file_ext	<string>	Default file extension for DEM force files		*.dem
dem_force_file_name	<string>	Default name for DEM force file		demfor
file_filter	<string>	Sets the default file filter used for PTF/D3PLOT files		*.ptf
file_names	<string>	Controls input filename syntax. LSTC = d3*, OASYS/ARUP = job.ptf*	OASYS, ARUP, LSTC	OASYS
file_skip	<integer>	Skips n missing d3plot/ptf files. LSTC = d3*, OASYS/ARUP = job.ptf*	0 - 999	50
fsi_force_file_ext	<string>	Default file extension for FSI force files		*.fff
fsi_force_file_name	<string>	Default name for FSI force file		fsifor
group_file_location	<string>	Directory for groups (jobname.grp) file (instead of job directory)		<none>
increment_fname	<logical>	Whether new filenames are	TRUE, FALSE	TRUE

		incremented by appending 001, 002, etc...		
input_warnings	<string>	Switching and location of warnings on input	NONE, DIALOGUE, MENU	DIALOGUE
master_group_file	<string>	Valid master (ascii) groups filename		<none>
mat_prop_location	<string>	Custom path for the mat_prop.csv file of material history variables		<none>
open_models_in_w1	<logical>	Controls if D3PLOT opens models in Window 1	TRUE, FALSE	FALSE
output_directory	<string>	Default output directory for images and other files when multiple models have been read	LATEST_MODEL_READ, FIRST_MODEL_READ	LATEST_MODEL_READ
primer_version	<string>	Name of PRIMER executable for ZTF file auto-creation		primer21_64.exe
read_asc_file	<logical>	Read ASC (ascii groups) file	TRUE, FALSE	TRUE
read_ctf_file	<logical>	Read CTF file	TRUE, FALSE	TRUE
read_lsda_file	<logical>	Read a LSDA (binout) file	TRUE, FALSE	TRUE
read_prp_file	<logical>	Read PRP (properties) file	TRUE, FALSE	TRUE
read_set_file	<logical>	Read SET (settings) file	TRUE, FALSE	TRUE
read_xtf_file	<logical>	Read XTF file	TRUE, FALSE	TRUE
read_ztf_file	<logical>	Read ZTF file	TRUE, FALSE	TRUE
ubd_file_dispose	<string>	Handling of UBIN data (.ubd) files on	LEAVE, DELETE	LEAVE

		model close and exit		
ubd_file_location	<string>	Optional alternative directory for UBIN data (.ubd) files		<none>

Font settings.

Preference	Type	Description	Valid arguments	Default
all_fonts	<string>	Graphics typeface and strength	HELVETICA, HELVETICA-BOLD, TIMES, TIMES-BOLD, COURIER, COURIER-BOLD, DEFAULT	DEFAULT
label_size	<string>	Font size for labels	8, 10, 12, 14, 18, 24, Default	Default
title_size	<string>	Font size for title	8, 10, 12, 14, 18, 24, Default	Default
clock_size	<string>	Font size for clock	8, 10, 12, 14, 18, 24, Default	Default
contour_size	<string>	Font size for contour bar	8, 10, 12, 14, 18, 24, Default	Default
graticule_size	<string>	Font size for graticule	8, 10, 12, 14, 18, 24, Default	Default
footer_size	<string>	Font size for footer	8, 10, 12, 14, 18, 24, Default	Default

General graphics initialisation and settings.

Preference	Type	Description	Valid arguments	Default
line_width	<integer>	Width of overlay/wireframe lines	1 - 10	1
line_antialias	<string>	Draw lines using antialiasing	OFF, ON	OFF
correct_beam_3rd_node	<logical>	Copy location of beam 3rd node from first state to undeformed	TRUE, FALSE	TRUE
draw_update	<string>	Draw update mode for the states slider	RELEASE, SLIDE	RELEASE
graphics_refresh	<string>	Refresh graphics window when exposed	OFF, ON	ON

graphics_type	<string>	Graphics format to start D3PLOT with	X8, X24, X, Opengl, Stereo, Default	Opengl
header_type	<string>	Header type	TITLE, FILENAME, DATABASE, DIRECTORY	TITLE
header_nchars	<integer>	Maximum number of characters to display in header		0
initial_plot_mode	<string>	Initial drawing mode	LINE, HIDDEN, SHADED	SHADED
initial_view_orientation	<string>	Initial view orientation for the graphics window.	+XY, +YZ, +XZ, +ISO, -XY, -YZ, -XZ, -ISO	+XY
intel_hd_use_shaders	<string>	Control usage of hardware shaders on Intel HD graphics cards	AUTO_DETECT, FORCE_OFF, FORCE_ON	AUTO_DETECT
batch_mode_use_shaders	<logical>	Use hardware shaders when running in 'batch' mode	TRUE, FALSE	TRUE
label_background	<string>	Label background display	ON, OFF	ON
max_frame_rate	<string>	Initial requested animation rate in frames/second	UNLIMITED, 1, 5, 10, 20, 25, 50, 75, 100	100
maximise	<logical>	Maximise window when D3PLOT started	TRUE, FALSE	TRUE
overlay_mode	<string>	Overlay drawn	OFF, FREE, FEATURE, ALL	FREE
placement	<string>	Location for initial window on multi-screen display	LEFT, RIGHT, BOTTOM, TOP, LEFT_BOTTOM, LEFT_TOP, RIGHT_BOTTOM, RIGHT_TOP	<none>
plot_border	<string>	Border drawn on plot	OFF, ON	OFF
plot_contour_bar	<string>	Contour bar drawn on plot	OFF, ON	ON
plot_clock	<string>	Clock drawn on plot	OFF, ON	ON
plot_date	<string>	Date drawn on plot	OFF, ON	OFF
plot_header	<string>	Header drawn on plot	OFF, ON	ON
plot_disp_mag	<string>	Magnifications drawn on plot	OFF, ON	ON
plot_triad	<string>	Triad drawn on plot	OFF, ON	ON
plot_d3plot_prefix	<string>	Controls whether D3PLOT prefix is used in graphics window title	OFF, ON	ON
pre_blank_all	<logical>	Blank everything before reading Reporter properties file	TRUE, FALSE	FALSE

rhs_number_columns	<integer>	Number of columns of Tools buttons	4 - 50	4

Graticule settings.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
graticule_number_format_type	<string>	Number format type for graticule	AUTO, SCIENTIFIC, GENERAL, MANUAL	AUTO
graticule_dec_places	<integer>	Number of decimal places to display on graticule	0 - 9	3
graticule_exponent	<integer>	Value of exponent to use on graticule	-99 - 99	3
graticule_line_colour	<string>	Graticule line colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLACK
graticule_plane_colour	<string>	Graticule plane colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE,	GREY

			DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	
graticule_text_colour	<string>	Graticule text colour (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLACK

The following options for groups. See Groups.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
use_first_entity_in_group	<logical>	Use first entity in group when writing properties to group ascii file	TRUE, FALSE	FALSE
graphical_user_interface				
gui_theme	<string>	Graphical User Interface (GUI) theme	LIGHT, DARK, CLASSIC, LEGACY	LIGHT

gui_styling_mode	<string>	Graphical User Interface (GUI) styling and decoration	NOT_USED, TIME_LIMIT, ALWAYS	TIME_LIMIT
gui_styling_tlimit	<integer>	Graphical User Interface (GUI) menu repaint time limit to turn off decorations	0 - 100000	500

The following options define the images, movies, 3D export, ray tracing, background and watermark images options. See watermarks section.

Preference	Type	Description	Valid arguments	Default
background_image	<string>	Valid background image filename		<none>
image_format	<string>	Default image format	BMP_8_C, BMP_8_UN, PNG_8, GIF_8, BMP_24_UN, PNG_24, JPG_24, PPM_24	PNG_24
movie_format	<string>	Default movie format	MP4, AVI, GIF	MP4
glb_format	<string>	Default 3D export format	COMPRESSED, UNCOMPRESSED	COMPRESSED
output_frame_rate	<integer>	Default animation rate in frames/second for writing movies		25
playback_method	<string>	Background movie playback method	STREAMED, CACHED	STREAMED
povray_executable	<string>	Executable for POV-Ray ray tracer executable		<none>
povray_matlib_file	<string>	Optional library file of locally defined POV-Ray material definitions		<none>
povray_rules_file	<string>	Rules file for automatic assignment of POV-Ray material definitions		<none>
watermark_image	<string>	Valid watermark image filename		<none>
white_background_image	<logical>	Write images with white background	TRUE, FALSE	FALSE

The following strings and values control the maximum number of labels that will be displayed.

Preference	Type	Description	Valid arguments	Default
max_labels	<integer>	Maximum number of labels to display	1 - 2147483646	1000
label_warning	<logical>	Display a warning if the maximum number of labels is reached	TRUE, FALSE	TRUE

label_picked_items	<logical>	Automatically label picked items	TRUE, FALSE	TRUE

The following strings and values control laser plotting setup (see laser plotting section for more on laser plotting).

Preference	Type	Description	Valid arguments	Default
laser_paper_size	<string>	Default paper size	US, A4	A4
laser_orientation	<string>	Default page orientation	Portrait, Landscape	Landscape
laser_mode	<string>	Default laser mode	Colour, Greyscale	Greyscale
laser_insert_file	<string>	Valid filename		<none>
laser_top_margin	<real>	Top margin size in mm		10
laser_bottom_margin	<real>	Bottom margin size in mm		30
laser_left_margin	<real>	Left margin size in mm		20
laser_right_margin	<real>	Right margin size in mm		10

Window layout commands control how multiple graphics windows are positioned and sized, and give some further options. (See window management section)

Preference	Type	Description	Valid arguments	Default
window_layout	<string>	Multiple window layout type	TILE_WIDE, TILE_TALL, CASCADE, 1x1, 2x2, 3x3	TILE_WIDE
auto_hide	<logical>	Hide graphics window function bar	TRUE, FALSE	FALSE
show_tabs	<logical>	Show window tabs on panels	TRUE, FALSE	TRUE
windows_same_size	<logical>	Windows initially the same size	TRUE, FALSE	FALSE
common_window_borders	<logical>	Dragging borders between adjacent windows moves the common border	TRUE, FALSE	TRUE
snap_window_position	<logical>	When dragging graphics windows they snap into slots in the window grid	TRUE, FALSE	TRUE

The following **`shaded_<xxx>`** options affect lighting (**SH**aded and **SI**) plots only.

Preference	Type	Description	Valid arguments	Default
shaded_ambient	<real>	Percentage ambient light (0-100)	0.0 - 100.0	40
shaded_diffuse	<real>	Percentage diffuse brightness (0-100)	0.0 - 100.0	90
shaded_shininess	<real>	Percentage specular brightness (0-100)	0.0 - 100.0	30

shaded_saturation	<real>	Percentage colour saturation (0-100)	0.0 - 100.0	50
shading_type	<string>	Shading type	FLAT, SMOOTH, DITHERED	FLAT

The following options affect the operation and appearance of linked T/HIS: (seeD3PLOT-T/HIS link section)

Preference	Type	Description	Valid arguments	Default
this_window_location	<string>	Linked T/HIS window location on screen	SIBLING, CHILD, DOCKED	SIBLING
this_link_version_64	<string>	Name of 64 bit executable for T/HIS link		this21_64.exe
this_link_timeout	<integer>	Timeout period (seconds) for T/HIS link	1 - 1000	5

The following options affect the appearance and behaviour of the graphical user interface (see customising the user interface section), left handed support, and the mouse

Preference	Type	Description	Valid arguments	Default
menu_layout_version	<integer>	Controls layout of right hand side toolbox and states menu)	14 - 15	15
display_factor	<real>	Factor on display size (0.5 - 2.0, automatic if undefined)	0.5 - 2.0	1.2
display_brightness	<real>	Menu brightness (0.0-1.0)	0.0 - 1.0	1.0
display_saturation	<real>	Menu colour saturation (0.0-1.0)	0.0 - 1.0	1.0
button_gradation	<real>	Button shade gradation (0.0-1.0)	0.0 - 1.0	0.0
dv_sync_windows	<string>	Dyn view method(s) for synchronising windows	ICON, ICON+CAPS, ICON+NUM, ICON+CAPS+NUM	ICON+CAPS
dv_left_shift	<string>	Dyn view action for shift + Left mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ROTATION_XYZ

dv_middle_shift	<string>	Dyn view action for shift + Middle mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	TRANSLATION
dv_right_shift	<string>	Dyn view action for shift + Right mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ZOOM_UP_+VE
dv_left_ctrl	<string>	Dyn view action for ctrl + Left mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ROTATION_XYZ
dv_middle_ctrl	<string>	Dyn view action for ctrl + Middle mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	TRANSLATION
dv_right_ctrl	<string>	Dyn view action for ctrl + Right mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ZOOM_UP_+VE
dv_left_both	<string>	Dyn view action for shift+ctrl + Left mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ROTATION_XYZ
dv_middle_both	<string>	Dyn view action for shift+ctrl + Middle mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION,	TRANSLATION

			ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	
dv_right_both	<string>	Dyn view action for shift+ctrl + Right mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ZOOM_UP_+VE
dv_shift_action	<string>	Dynamic viewing mode for shift + mouse button	CURRENT, WIREFRAME, FREE_EDGE, UNUSED	CURRENT
dv_ctrl_action	<string>	Dynamic viewing mode for ctrl + mouse button	CURRENT, WIREFRAME, FREE_EDGE, UNUSED	WIREFRAME
dv_both_action	<string>	Dynamic viewing mode for shift+ctrl + mouse button	CURRENT, WIREFRAME, FREE_EDGE, UNUSED	FREE_EDGE
font_cache	<logical>	Whether to use cached fonts on Linux machines with no core X11 fonts loaded	TRUE, FALSE	TRUE
font_quality	<string>	The quality of font rendering in the graphical user interface	PLAIN, ANTI_ALIAS	ANTI_ALIAS
font_scaling	<string>	Whether text in GUI buttons can be scaled down to fit (TRUE means both width and height)	FALSE, WIDTH, HEIGHT, TRUE	WIDTH
font_silent	<logical>	whether to write explanatory text if wanted fonts are not found	TRUE, FALSE	FALSE
font_size	<string>	Menu font size	TINY, SMALL, DEFAULT, LARGE, HUGE	DEFAULT
font_type	<string>	Menu font typeface and strength	HELVETICA, HELVETICA-BOLD, TIMES, TIMES-BOLD, COURIER, COURIER-BOLD	HELVETICA
unix_prop_font	<string>	GUI proportional font for menu		Helvetica

		panels on Linux/Unix		
unix_mono_font	<string>	GUI monospaced font for listing boxes on Linux/Unix		Courier New
windows_prop_font	<string>	GUI proportional font for menu panels on Windows		Helvetica
windows_mono_font	<string>	GUI monospaced font for listing boxes on Windows		Courier New
left_handed	<string>	Left handed switching of mouse and/or keyboard	NONE, MOUSE, KEYBOARD, ALL	NONE
max_comp_popup_rows	<integer>	Number of rows in the component selection popups before a scrollbar is added	2 - 100	25
zoom_factor	<real>	Zoom Factor for mouse wheel (0.01-1.0)	0.01 - 1.0	0.05
czoom_factor	<real>	Factor for right mouse dynamic zoom (0.01-0.2)	0.01 - 0.2	0.05
kzoom_factor	<real>	Factor for +/- keyboard short-cut keys	0.01 - 100.0	2.0
menu_dragging_mode	<string>	Mode used when moving menu panels with the mouse	WIREFRAME, OPAQUE	WIREFRAME
menu_resize	<string>	Which panel borders are free to resize	NONE, L, R, T, B, LR, LT, LB, LRT, LRTB, RT, RB, RTB, TB, ALL	ALL
mouse_3d_rotation_factor	<real>	Factor applied to the speed of rotation when using a 3D mouse		1.0
mouse_3d_pan_factor	<real>	Factor applied to the speed of panning when using a 3D mouse		1.0
mouse_3d_zoom_factor	<real>	Factor applied to the speed of zooming when using a 3D mouse		1.0

mouse_action_middle_button	<string>	Set the action for the middle mouse key during picking	APPLY, REJECT, DESELECT	REJECT
mouse_action_right_button	<string>	Set the action for the right mouse key during picking	APPLY, REJECT, DESELECT	DESELECT
menus_and_picking				
menu_sketch	<string>	Whether or not to show sketch menu items when cursor hovered over menu row	OFF, ON	ON
menu_label	<string>	Whether or not menu sketching also shows item labels	OFF, ON	ON
predictive_pick	<string>	Whether or not to show what will be picked based on the current cursor position	OFF, ON	ON
predictive_label	<string>	Whether or not predictive picking also shows item labels	OFF, ON	ON
query_ambiguous	<string>	If screen picking is ambiguous, ON will offer the selection menu, OFF will select nearest	OFF, ON	ON
show_qpick_stack_warning	<logical>	Display Quick Pick memory stack size warning and option to clear stack if threshold is hit.	TRUE, FALSE	TRUE

If a selection menu is not wide enough to display all the contents, it can be expanded automatically by the following (see menu auto expansion section)

Preference	**Type**	**Description**	**Valid arguments**	**Default**
menu_expand	<string>	Automatic menu expansion on/off switch	OFF, ON, EXPAND, UNDOCK	ON
menu_expand_delay	<real>	Factor on delay time before expansion	0.1 - 5.0	1.0
menu_expand_speed	<real>	Factor on menu expansion speed	0.1 - 5.0	1.0

Controls options for the measure menu

Preference	Type	Description	Valid arguments	Default
measure_auto_create	<logical>	If measure auto-create is active when the measure menu is first loaded	TRUE, FALSE	TRUE
measure_dialogue_format_vec_rows	<logical>	Vectors in the dialogue window are formatted in the common way with vectors written in rows	TRUE, FALSE	FALSE

The following control the treatment of results for solid elements with multiple integration points

Preference	Type	Description	Valid arguments	Default
d3plot_solid_8	<logical>	Average multiple integration point results on read	TRUE, FALSE	FALSE
multi_ip_mode	<string>	How to treat multiple integration point results in the absence of element formulation from ztf	ave_all_ips, max_all_ips, first_ip	ave_all_ips

Options to control the reading of Nastran OP2 and Eigenvector analysis files. As of v21, this panel can also magnify the mode shapes of other eigenvalue analysis files (e.g. d3eigv). New preferences (those prefixed with eigmode_) are provided to reflect this while the old preferences (prefixed with nas_) are retained for backwards compatibility. If both an eigmode_ and the corresponding nas_ preferences are present the eigmode_ preference value will be used.

Preference	Type	Description	Valid arguments	Default
nas_disp_factor_type	<string>	How to scale Nastran displacements	ABSOLUTE, PERCENT	PERCENT
nas_abs_disp_factor	<real>	Absolute displacement scale factor		1.0
nas_pct_disp_factor	<real>	Percentage displacement scale factor		15.0
eigmode_disp_factor_type	<string>	How to scale Nastran and d3eigv displacements	ABSOLUTE, PERCENT	PERCENT
eigmode_abs_disp_factor	<real>	Absolute displacement scale factor		1.0
eigmode_pct_disp_factor	<real>	Percentage displacement scale factor		15.0

The following control settings related to object menus

Preference	Type	Description	Valid arguments	Default
vis_menu_position	<string>	Position of the Vis menu - graphics area or docked to parent	FLOATING, DOCKED	DOCKED

The following control settings related to quickfind

Preference	Type	Description	Valid arguments	Default
quickfind_unmatched_text_colour	<string>	Text colour for unmatched characters (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLACK
quickfind_matched_text_colour	<string>	Text colour for matched characters (hex code e.g. 0XA1B2C3 or core colour name e.g. OLIVE)	WHITE, BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, RED/MAGENTA, GREEN/CYAN, INDIGO, YELLOW/GREEN, LIGHT_BLUE, DARK_ORANGE, PALE_YELLOW, GOLD, OLIVE, DARK_MAGENTA, MEDIUM_GREEN, CYAN/BLUE, HOT_PINK, RED/ORANGE, SEA_GREEN, MAROON, DARK_GREEN, PURPLE, NAVY, DARK_GREY, GREY, LIGHT_GREY	BLUE
quickfind_found_list_length	<integer>	Number of items to display in the found list	1 - 20	10

quickfind_recent_history	<integer>	Number of recently selected items to store	0 - 2147483646	10
quickfind_sequential_bonus	<integer>	Modifier for two successful adjacent matches	1 - 100	50
quickfind_word_start_bonus	<integer>	Modifier for successful match at word beginning	1 - 100	10
quickfind_box_size	<string>	Size and layout of Search box	SMALL, LARGE	SMALL

Options to control the behaviour and appearance of the Part Tree.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
ptree_parts_top_level	<logical>	If TRUE parts are always expanded at the top level	TRUE, FALSE	TRUE
ptree_show_beam	<logical>	If TRUE a Beam category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_group	<logical>	If TRUE a Groups category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_joint	<logical>	If TRUE a Joints category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_mass	<logical>	If TRUE a Mass category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_pretensioner	<logical>	If TRUE a Pretensioner category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_retractor	<logical>	If TRUE a Retractor category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_seatbelt	<logical>	If TRUE a Seatbelt category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_segment	<logical>	If TRUE a (contact) Segment category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_shell	<logical>	If TRUE a Shell category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_slipring	<logical>	If TRUE a Slipring category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_solid	<logical>	If TRUE a Solid category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_spring	<logical>	If TRUE a Spring category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_surface	<logical>	If TRUE a (contact) Surface category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_tshell	<logical>	If TRUE a Thick Shell category will be included in the tree	TRUE, FALSE	FALSE
ptree_show_wall	<logical>	If TRUE a Wall category will be included in the tree	TRUE, FALSE	FALSE

The following options define how Javascripts are processed by D3Plot. See the JavaScript API reference manual for further details.

Preference	Type	Description	Valid arguments	Default
modules_directory	<string>	Directory for D3PLOT to look for modules in		<none>
script_directory	<string>	Directory in which D3PLOT looks for scripts		$OA_INSTALL/d3plot_library/scripts
javascript_memory_size	<integer>	Maximum memory allocated for garbage collection		25

The following options define how D3PLOT sessions are processed See UTILITIES-SESSION_FILE section for further details.

Preference	Type	Description	Valid arguments	Default
session_auto_save	<string>	Save a session unconditionally on exit	OFF, ON	OFF
session_save_option	<string>	Location for automatically saving sessions	HOME, USER_DEFINED, DESKTOP	HOME
session_save_dir	<string>	User-defined location for session save		<none>

Keys can have functions assigned to them:

Preference	Type	Description	Valid arguments	Default
F1_key	<string>	Shortcut for F1		<none>
F2_key	<string>	Shortcut for F2		<none>
F3_key	<string>	Shortcut for F3		<none>
F4_key	<string>	Shortcut for F4		<none>
F5_key	<string>	Shortcut for F5		<none>
F6_key	<string>	Shortcut for F6		<none>
F7_key	<string>	Shortcut for F7		<none>
F8_key	<string>	Shortcut for F8		<none>
F9_key	<string>	Shortcut for F9		<none>
F10_key	<string>	Shortcut for F10		<none>
F11_key	<string>	Shortcut for F11		<none>
F12_key	<string>	Shortcut for F12		<none>
A_key	<string>	Shortcut for A		AUTOSCALE
B_key	<string>	Shortcut for B		BLANK

C_key	<string>	Shortcut for C		TIDY_MENUS
D_key	<string>	Shortcut for D		DRAG_CUT
E_key	<string>	Shortcut for E		EXPORT_MENU
F_key	<string>	Shortcut for F		FRINGE
G_key	<string>	Shortcut for G		NEW_WINDOW
H_key	<string>	Shortcut for H		HIDDEN
I_key	<string>	Shortcut for I		ICONISE
J_key	<string>	Shortcut for J		TOGGLE_PICK_LABEL
K_key	<string>	Shortcut for K		RESET_VIS
L_key	<string>	Shortcut for L		LINE
M_key	<string>	Shortcut for M		MEASURE
N_key	<string>	Shortcut for N		CUT_PLANE
O_key	<string>	Shortcut for O		DISPLAY
P_key	<string>	Shortcut for P		TOGGLE_GLOBAL_PP
Q_key	<string>	Shortcut for Q		QUICK_PICK
R_key	<string>	Shortcut for R		REVERSE
S_key	<string>	Shortcut for S		SHADED
T_key	<string>	Shortcut for T		TIDY_MENUS
U_key	<string>	Shortcut for U		UNBLANK
V_key	<string>	Shortcut for V		VIEW_MENU
W_key	<string>	Shortcut for W		IMAGE_WRITE
X_key	<string>	Shortcut for X		CUT_SECTION
Y_key	<string>	Shortcut for Y		CYCLE_OVERLAY
Z_key	<string>	Shortcut for Z		ZOOM
a_key	<string>	Shortcut for a		AUTOSCALE
b_key	<string>	Shortcut for b		BLANK
c_key	<string>	Shortcut for c		TIDY_MENUS
d_key	<string>	Shortcut for d		DRAG_CUT
e_key	<string>	Shortcut for e		ENTITIES
f_key	<string>	Shortcut for f		FRINGE
g_key	<string>	Shortcut for g		NEW_WINDOW
h_key	<string>	Shortcut for h		HIDDEN
i_key	<string>	Shortcut for i		ICONISE
j_key	<string>	Shortcut for j		TOGGLE_PICK_LABEL
k_key	<string>	Shortcut for k		RESET_VIS
l_key	<string>	Shortcut for l		LINE
m_key	<string>	Shortcut for m		MEASURE
n_key	<string>	Shortcut for n		CUT_PLANE
o_key	<string>	Shortcut for o		DISPLAY
p_key	<string>	Shortcut for p		TOGGLE_CURR_PP
q_key	<string>	Shortcut for q		QUICK_PICK
r_key	<string>	Shortcut for r		REVERSE
s_key	<string>	Shortcut for s		SHADED
t_key	<string>	Shortcut for t		TIDY_MENUS
u_key	<string>	Shortcut for u		UNBLANK
v_key	<string>	Shortcut for v		VIEW_MENU

w_key	<string>	Shortcut for w		IMAGE_WRITE
x_key	<string>	Shortcut for x		CUT_SECTION
y_key	<string>	Shortcut for y		CYCLE_OVERLAY
z_key	<string>	Shortcut for z		ZOOM
SPACE_key	<string>	Shortcut for space		ANIMATE
ONE_key	<string>	Shortcut for 1		VIEW_P_XY
TWO_key	<string>	Shortcut for 2		VIEW_P_YZ
THREE_key	<string>	Shortcut for 3		VIEW_P_XZ
FOUR_key	<string>	Shortcut for 4		VIEW_P_ISO
FIVE_key	<string>	Shortcut for 5		VIEW_N_XY
SIX_key	<string>	Shortcut for 6		VIEW_N_YZ
SEVEN_key	<string>	Shortcut for 7		VIEW_N_XZ
EIGHT_key	<string>	Shortcut for 8		VIEW_N_ISO
NINE_key	<string>	Shortcut for 9		<none>
ZERO_key	<string>	Shortcut for 0		EXPORT
EXCLAMATION_key	<string>	Shortcut for !		<none>
DOUBLEQUOTE_key	<string>	Shortcut for "		<none>
HASH_key	<string>	Shortcut for #		<none>
DOLLAR_key	<string>	Shortcut for $		<none>
PERCENT_key	<string>	Shortcut for %		<none>
AMPERSAND_key	<string>	Shortcut for &		<none>
SINGLEQUOTE_key	<string>	Shortcut for '		<none>
LEFTBRACKET_key	<string>	Shortcut for (		<none>
RIGHTBRACKET_key	<string>	Shortcut for)		<none>
ASTERISK_key	<string>	Shortcut for *		<none>
PLUS_key	<string>	Shortcut for +		ZOOM_IN
COMMA_key	<string>	Shortcut for ,		<none>
MINUS_key	<string>	Shortcut for -		ZOOM_OUT
DOT_key	<string>	Shortcut for .		<none>
SLASH_key	<string>	Shortcut for /		SHORTCUT
COLON_key	<string>	Shortcut for :		<none>
SEMICOLON_key	<string>	Shortcut for ;		<none>
LESSTHAN_key	<string>	Shortcut for <		<none>
EQUALS_key	<string>	Shortcut for =		ZOOM_IN
GREATERTHAN_key	<string>	Shortcut for >		<none>
QUESTIONMARK_key	<string>	Shortcut for ?		SHORTCUT
AT_key	<string>	Shortcut for @		<none>
LEFTSQUAREBRACKET_key	<string>	Shortcut for [		<none>
BACKSLASH_key	<string>	Shortcut for \		<none>
RIGHTSQUAREBRACKET_key	<string>	Shortcut for]		<none>
CIRCUMFLEX_key	<string>	Shortcut for ^		<none>
UNDERSCORE_key	<string>	Shortcut for _		ZOOM_OUT
BACKTICK_key	<string>	Shortcut for `		<none>
LEFTCURLYBRACKET_key	<string>	Shortcut for {		<none>
PIPE_key	<string>	Shortcut for |		<none>
RIGHTCURLYBRACKET_key	<string>	Shortcut for }		<none>

TILDE_key	<string>	Shortcut for ~		<none>
SM_BUTTON1_key	<string>	Shortcut for 3D SpaceMouse Button 1		VIEW_P_XY
SM_BUTTON2_key	<string>	Shortcut for 3D SpaceMouse Button 2		VIEW_N_XZ
SM_BUTTON3_key	<string>	Shortcut for 3D SpaceMouse Button 3		VIEW_P_XZ
SM_BUTTON4_key	<string>	Shortcut for 3D SpaceMouse Button 4		VIEW_P_YZ
SM_BUTTON5_key	<string>	Shortcut for 3D SpaceMouse Button 5		<none>
SM_BUTTON6_key	<string>	Shortcut for 3D SpaceMouse Button 6		<none>
SM_BUTTON7_key	<string>	Shortcut for 3D SpaceMouse Button 7		<none>
SM_BUTTON8_key	<string>	Shortcut for 3D SpaceMouse Button 8		<none>
SM_BUTTON9_key	<string>	Shortcut for 3D SpaceMouse Button 9		<none>
SM_BUTTON10_key	<string>	Shortcut for 3D SpaceMouse Button 10		<none>
SM_BUTTON11_key	<string>	Shortcut for 3D SpaceMouse Button 11		<none>
SM_BUTTON12_key	<string>	Shortcut for 3D SpaceMouse Button 12		<none>
SM_BUTTON13_key	<string>	Shortcut for 3D SpaceMouse Button 13		<none>
SM_BUTTON14_key	<string>	Shortcut for 3D SpaceMouse Button 14		<none>
SM_BUTTON15_key	<string>	Shortcut for 3D SpaceMouse Button 15		<none>
SM_BUTTON16_key	<string>	Shortcut for 3D SpaceMouse Button 16		<none>

SM_BUTTON17_key	<string>	Shortcut for 3D SpaceMouse Button 17		<none>
SM_BUTTON18_key	<string>	Shortcut for 3D SpaceMouse Button 18		<none>
SM_BUTTON19_key	<string>	Shortcut for 3D SpaceMouse Button 19		<none>
SM_BUTTON20_key	<string>	Shortcut for 3D SpaceMouse Button 20		<none>
SM_BUTTON21_key	<string>	Shortcut for 3D SpaceMouse Button 21		<none>
SM_BUTTON22_key	<string>	Shortcut for 3D SpaceMouse Button 22		<none>
SM_BUTTON23_key	<string>	Shortcut for 3D SpaceMouse Button 23		<none>
SM_BUTTON24_key	<string>	Shortcut for 3D SpaceMouse Button 24		<none>
SM_BUTTON25_key	<string>	Shortcut for 3D SpaceMouse Button 25		<none>
SM_BUTTON26_key	<string>	Shortcut for 3D SpaceMouse Button 26		<none>
SM_BUTTON27_key	<string>	Shortcut for 3D SpaceMouse Button 27		<none>
SM_BUTTON28_key	<string>	Shortcut for 3D SpaceMouse Button 28		<none>
SM_BUTTON29_key	<string>	Shortcut for 3D SpaceMouse Button 29		<none>
SM_APPLICATION_key	<string>	Shortcut for 3D SpaceMouse Application Button		SHORTCUT_3D
SM_FIT_key	<string>	Shortcut for 3D SpaceMouse Fit Button		AUTOSCALE

The following options allow the user to change the symbols (and their quality) representing various entities which are drawn in plots. See DISPLAY_OPTIONS for further details.

Preference	**Type**	**Description**	**Valid arguments**	**Default**

abp_symbol	<string>	Symbol for Airbag Particles	POINT, CUBE, SPHERE	SPHERE
abp_quality	<integer>	Quality of Airbag Particle sphere symbol	1 - 5	1
abp_leakage_in_bag	<string>	Airbag Particle leakage in bag selected	ON, OFF	ON
abp_leakage_porosity	<string>	Airbag Particle leakage porosity selected	ON, OFF	ON
abp_leakage_vented	<string>	Airbag Particle leakage vented selected	ON, OFF	ON
abp_leakage_mpp_error	<string>	Airbag Particle leakage mpp error selected	ON, OFF	OFF
des_symbol	<string>	Symbol for Discrete Sphere elements	POINT, CUBE, SPHERE	SPHERE
des_quality	<integer>	Quality of Discrete Sphere symbol	1 - 5	2
beam_line_thickness	<real>	Line thickness of beams if thick lines are on	0.0 - 10000000.0	<none>
discrete_beam_radius	<real>	Radius of discrete beams if true beam sections are on	0.0 - 10000000.0	<none>
sph_symbol	<string>	Symbol for SPH elements	POINT, CUBE, SPHERE	SPHERE
sph_quality	<integer>	Quality of SPH sphere symbol	1 - 5	2
swld_symbol	<string>	Symbol for type for Spotwelds	DEFAULT, SPHERE, BEAM	DEFAULT
swld_quality	<integer>	Quality of Spotweld sphere symbol	1 - 5	2
swld_radius	<string>	Display spotwelds using the PANEL gap, TRUE radius or a FIXED radius	PANEL, TRUE, FIXED	PANEL
swld_panel_factor	<real>	Factor to mulitple PANEL gap by when drawing spotwelds spheres	0.0 - 10000000.0	1.5
swld_true_factor	<real>	Factor to mulitple TRUE radius by when drawing spotwelds spheres	0.0 - 10000000.0	1.0
swld_fixed_size	<real>	Default radius used when drawing spotwelds with a FIXED radius	0.0 - 10000000.0	1.0
swld_scale_by_value	<logical>	TRUE if spotweld radius is going to be scaled by the value	TRUE, FALSE	FALSE
spring_width	<integer>	Thickness (pixels) used to draw springs	1 - 10	2
true_beam_sections	<string>	Method of drawing beam elements (deprecated: TRUE = TRUE_SECTION, FALSE = PLAIN_LINE)	PLAIN_LINE, THICK_LINE, TRUE_SECTION, TRUE, FALSE	PLAIN_LINE
beam_spwld_end_caps	<string>	Beam spotweld end caps added	ON, OFF	OFF
spg_mode	<string>	Default drawing mode for SPG Solid Parts	SOLID, SPHERE	SPHERE

The following settings allow high performance graphics settings to be tuned. It is recommended that you do not modify these in the preferences editor, but rather use the TUNE option and then SAVE_SETTINGS.

Preference	Type	Description	Valid arguments	Default
gtune_varray	<integer>	Whether or not to use vertex arrays	0 - 2	0
gtune_vbo_verts	<integer>	Whether or not to use VBOs for vertices	0 - 2	0
gtune_vbo_coords	<integer>	Whether or not to use VBOs for coordinates	0 - 2	0
gtune_vbo_limit	<integer>	How VBO usage is limited (explicit size in MBytes, or -1 for auto)	-1 - 1048576	-1
gtune_shader	<integer>	Whether or not to use shaders	0 - 2	0
gtune_mbr	<integer>	Whether or not to use the MBR extension for VBOs	0 - 3	0

The following options control automatic mesh coarsening, and set a model size threshold at which this is implemented. This can be useful if you habitually process very large models and are happy to accept slightly poorer image quality in return for faster graphics.

Preference	Type	Description	Valid arguments	Default
auto_coarsen	<string>	Automatic mesh coarsening on initial read	OFF, MILD, SEVERE	OFF
coarsen_threshold	<integer>	Min number of shells for auto-coarsening	0 - 1000000000	250000

The following options control threading.

Preference	Type	Description	Valid arguments	Default
threading_active	<logical>	Whether or not to use threading	TRUE, FALSE	TRUE
thread_contour_scan	<logical>	Whether or not to use threading for contour scanning	TRUE, FALSE	TRUE
thread_data_averaging	<logical>	Whether or not to use threading for data averaging at nodes	TRUE, FALSE	TRUE
thread_shell_normals	<logical>	Whether or not to use threading for shell normal calculations	TRUE, FALSE	TRUE
thread_shell_thickness	<logical>	Whether or not to use threading for shell thickness calculations	TRUE, FALSE	TRUE
thread_internal_faces	<logical>	Whether or not to use threading for internal face calculations	TRUE, FALSE	TRUE
thread_mark_visible	<logical>	Whether or not to use threading for marking visible items	TRUE, FALSE	TRUE
thread_cut_section	<logical>	Whether or not to use threading for cut section calculations	TRUE, FALSE	TRUE

thread_propagation	<logical>	Whether or not to use threading for flag propagation	TRUE, FALSE	TRUE
thread_femzip_data	<logical>	Whether or not to use threading for femzip data read	TRUE, FALSE	TRUE

The following options control disk_io.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
frag_local_size	<integer>	Buffer size (MBytes) reading fragmented data from local disk	0 - 2047	0
frag_network_size	<integer>	Buffer size (MBytes) reading fragmented data from network disk	0 - 2047	4
raw_network_size	<integer>	Buffer size (MBytes) reading raw data from network disk	0 - 2047	0
file_io_method	<string>	Local+Network file i/o method: raw+raw, raw+stream or stream+stream		raw+raw

The following control treatment of unicode

Preference	**Type**	**Description**	**Valid arguments**	**Default**
cjk_unix_font	<string>	Font to use for CJK text on unix machines		-misc-fixed-medium-r-normal-*-12-*-*-*-*-*-*-*
cjk_windows_font	<string>	Font to use for CJK text on windows machines		MS Gothic 10
file_encoding	<string>	Character encoding for script files	Latin-1, BIG5, EUC-CN, EUC-JP, EUC-KR, GB, GBK, ISO-2022-CN, ISO-2022-CN-EXT, ISO-2022-JP, ISO-2022-JP-2, ISO-2022-KR, JOHAB, Shift-JIS, UTF-8, UTF-16BE, UTF-16LE, UTF-16, UTF-32BE, UTF-32LE, UTF-32	Latin-1

The following `<xxx>_visibility` flags set the relevant **ENTITY** switches, and may subsequently be turned on/off manually in the normal way. The setting given here becomes the default for <reset> operations.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
mass_visibility	<string>	Lumped mass visibility	OFF, ON	OFF
spring_visibility	<string>	Spring/damper visibility	OFF, ON	ON
sbelt_visibility	<string>	Seatbelts etc. visibility	OFF, ON	ON
joint_visibility	<string>	Joint visibility	OFF, ON	ON
stonewall_visibility	<string>	Rigidwall visibility	OFF, ON	OFF
particle_visibility	<string>	Airbag particle visibility	OFF, ON	ON
connection_visibility	<string>	Connection visibility	OFF, ON	ON
section_visibility	<string>	Database X-Sect visibility	OFF, ON	ON
segment_visibility	<string>	Contact segment visibility	OFF, ON	OFF
segment_hatching	<string>	Contact segment hatching	OFF, ON	ON
spc_visibility	<string>	SPC visibility	OFF, ON	OFF
nrb_visibility	<string>	NRB visibility	OFF, ON	ON
loadpath_visibility	<string>	LoadPath visibility	OFF, ON	ON

The drive mappings allow D3PLOT to convert equivalent folder names from Windows to Unix and visa versa. This is currently only in use for the JavaScript function DriveMapFilename.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
drive_a	<string>	Mapping from Windows drive A: to unix path		<none>
drive_b	<string>	Mapping from Windows drive B: to unix path		<none>
drive_c	<string>	Mapping from Windows drive C: to unix path		<none>
drive_d	<string>	Mapping from Windows drive D: to unix path		<none>
drive_e	<string>	Mapping from Windows drive E: to unix path		<none>
drive_f	<string>	Mapping from Windows drive F: to unix path		<none>
drive_g	<string>	Mapping from Windows drive G: to unix path		<none>
drive_h	<string>	Mapping from Windows drive H: to unix path		<none>
drive_i	<string>	Mapping from Windows drive I: to unix path		<none>
drive_j	<string>	Mapping from Windows drive J: to unix path		<none>
drive_k	<string>	Mapping from Windows drive K: to unix path		<none>

drive_l	<string>	Mapping from Windows drive L: to unix path		<none>
drive_m	<string>	Mapping from Windows drive M: to unix path		<none>
drive_n	<string>	Mapping from Windows drive N: to unix path		<none>
drive_o	<string>	Mapping from Windows drive O: to unix path		<none>
drive_p	<string>	Mapping from Windows drive P: to unix path		<none>
drive_q	<string>	Mapping from Windows drive Q: to unix path		<none>
drive_r	<string>	Mapping from Windows drive R: to unix path		<none>
drive_s	<string>	Mapping from Windows drive S: to unix path		<none>
drive_t	<string>	Mapping from Windows drive T: to unix path		<none>
drive_u	<string>	Mapping from Windows drive U: to unix path		<none>
drive_v	<string>	Mapping from Windows drive V: to unix path		<none>
drive_w	<string>	Mapping from Windows drive W: to unix path		<none>
drive_x	<string>	Mapping from Windows drive X: to unix path		<none>
drive_y	<string>	Mapping from Windows drive Y: to unix path		<none>
drive_z	<string>	Mapping from Windows drive Z: to unix path		<none>

"oa_pref" arguments valid for all programs

Preference	**Type**	**Description**	**Valid arguments**	**Default**
file_names	<string>	Controls input filename syntax. LSTC = d3*, OASYS = job.ptf*	OASYS, LSTC	OASYS
html_application	<string>	Location of HTML browser		<none>
html_application_linux	<string>	Location of HTML browser for linux (use if the same oa_pref file is used for		<none>

		windows and linux)		
html_application_windows	<string>	Location of HTML browser for windows (use if the same oa_pref file is used for windows and linux)		<none>
image_format	<string>	Default image format	BMP_8_C, BMP_8_UN, PNG_8, GIF_8, BMP_24_UN, PNG_24, JPG_24, PPM_24	PNG_24
intel_hd_use_shaders	<string>	Control usage of hardware shaders on Intel HD graphics cards	AUTO_DETECT, FORCE_OFF, FORCE_ON	AUTO_DETECT
locale	<string>	Language and country locale to use (overrides system one)		<none>
manuals_url	<string>	URL of the online manuals		help.oasys-software.com/articles/?readerUiPreview=1#!
maximise	<logical>	Maximise window when Program is started	TRUE, FALSE	TRUE
online_manuals	<logical>	Open the online version of the manuals from Help buttons (TRUE) or open the local (offline) HTML copy (FALSE)	TRUE, FALSE	TRUE
pdf_application	<string>	Location of PDF browser		<none>
pdf_application_linux	<string>	Location of PDF browser for linux (use if the same oa_pref file is used for windows and linux)		<none>

pdf_application_windows	<string>	Location of PDF browser for windows (use if the same oa_pref file is used for windows and linux)		<none>
placement	<string>	Location for initial window on multi-screen display	LEFT, RIGHT, BOTTOM, TOP, LEFT_BOTTOM, LEFT_TOP, RIGHT_BOTTOM, RIGHT_TOP	<none>
start_in	<string>	Directory to start Program in		<none>
temp_file_expiry	<integer>	Age in days after which a temporary filename can be reused, 0 = never	0 - 10000	31
show_license_warning	<logical>	Display Window containing License System messages	TRUE, FALSE	TRUE
post_uses_primer	<logical>	ADMIN/INSTALL pref which allows D3Plot, T/his to take an available Primer license	TRUE, FALSE	TRUE
save_window_positions	<logical>	Save position of undocked windows between sessions	TRUE, FALSE	TRUE

The following control directories

Preference	**Type**	**Description**	**Valid arguments**	**Default**
home_dir	<string>	"home" directory for user		<none>
manuals_dir	<string>	Directory user manuals are installed in		<none>
temp_dir	<string>	temporary directory for user		<none>

write_checkpoint_files	<logical>	Record checkpoint files for the PRIMER, D3PLOT or T/His sessions.	TRUE, FALSE	FALSE
checkpoint_dir	<string>	Directory for checkpoint files. If omitted use cwd.		<none>
show_checkpoint_files	<logical>	Show checkpoint playback panel upon PRIMER, D3PLOT or T/His startup.	TRUE, FALSE	FALSE
graphics				
initial_view_orientation	<string>	Initial view orientation for the graphics window.	+XY, +YZ, +XZ, +ISO, -XY, -YZ, -XZ, -ISO	+XY
graphical_user_interface				
gui_theme	<string>	Graphical User Interface (GUI) theme	LIGHT, DARK, CLASSIC, LEGACY	LIGHT
gui_styling_mode	<string>	Graphical User Interface (GUI) styling and decoration	NOT_USED, TIME_LIMIT, ALWAYS	TIME_LIMIT
gui_styling_tlimit	<integer>	Graphical User Interface (GUI) menu repaint time limit to turn off decorations	0 - 100000	500

The following control laser options

Preference	Type	Description	Valid arguments	Default
laser_paper_size	<string>	Default paper size	US, A4	A4
laser_orientation	<string>	Default page orientation	Portrait, Landscape	Landscape
laser_top_margin	<real>	Top margin size in mm		10
laser_bottom_margin	<real>	Bottom margin size in mm		30
laser_left_margin	<real>	Left margin size in mm		20
laser_right_margin	<real>	Right margin size in mm		10

The following control menu and mouse attributes

Preference	Type	Description	Valid arguments	Default
display_factor	<real>	Factor on display size (0.5 - 2.0, automatic if undefined)	0.5 - 2.0	1.2
display_brightness	<real>	Menu brightness (0.0-1.0)	0.0 - 1.0	1.0

display_saturation	<real>	Menu colour saturation (0.0-1.0)	0.0 - 1.0	1.0
button_gradation	<real>	Button shade gradation (0.0-1.0)	0.0 - 1.0	0.0
dv_sync_windows	<string>	Dyn view method(s) for synchronising windows	ICON, ICON+CAPS, ICON+NUM, ICON+CAPS+NUM	ICON+CAPS
dv_left_shift	<string>	Dyn view action for shift + Left mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ROTATION_XYZ
dv_middle_shift	<string>	Dyn view action for shift + Middle mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	TRANSLATION
dv_right_shift	<string>	Dyn view action for shift + Right mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ZOOM_UP_+VE
dv_left_ctrl	<string>	Dyn view action for ctrl + Left mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ROTATION_XYZ
dv_middle_ctrl	<string>	Dyn view action for ctrl + Middle mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	TRANSLATION
dv_right_ctrl	<string>	Dyn view action for ctrl + Right mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z,	ZOOM_UP_+VE

			ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	
dv_left_both	<string>	Dyn view action for shift+ctrl + Left mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ROTATION_XYZ
dv_middle_both	<string>	Dyn view action for shift+ctrl + Middle mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	TRANSLATION
dv_right_both	<string>	Dyn view action for shift+ctrl + Right mouse	ROTATION_XYZ, ROTATION_XY, ROTATION_Z, ROTATION_SPHERE, TRANSLATION, ZOOM_UP_+VE, ZOOM_DOWN_+VE, UNUSED	ZOOM_UP_+VE
dv_shift_action	<string>	Dynamic viewing mode for shift + mouse button	CURRENT, WIREFRAME, FREE_EDGE, UNUSED	CURRENT
dv_ctrl_action	<string>	Dynamic viewing mode for ctrl + mouse button	CURRENT, WIREFRAME, FREE_EDGE, UNUSED	WIREFRAME
dv_both_action	<string>	Dynamic viewing mode for shift+ctrl + mouse button	CURRENT, WIREFRAME, FREE_EDGE, UNUSED	FREE_EDGE
font_cache	<logical>	Whether to use cached fonts on Linux machines with no core X11 fonts loaded	TRUE, FALSE	TRUE
font_quality	<string>	The quality of font rendering in the graphical user interface	PLAIN, ANTI_ALIAS	ANTI_ALIAS
font_scaling	<string>	Whether text in GUI buttons can	FALSE, WIDTH, HEIGHT, TRUE	WIDTH

		be scaled down to fit (TRUE means both width and height)		
font_silent	<logical>	whether to write explanatory text if wanted fonts are not found	TRUE, FALSE	FALSE
font_size	<string>	Menu font size	TINY, SMALL, DEFAULT, LARGE, HUGE	DEFAULT
font_type	<string>	Menu font typeface and strength	HELVETICA, HELVETICA-BOLD, TIMES, TIMES-BOLD, COURIER, COURIER-BOLD	HELVETICA
unix_prop_font	<string>	GUI proportional font for menu panels on Linux/Unix		Helvetica
unix_mono_font	<string>	GUI monospaced font for listing boxes on Linux/Unix		Courier New
windows_prop_font	<string>	GUI proportional font for menu panels on Windows		Helvetica
windows_mono_font	<string>	GUI monospaced font for listing boxes on Windows		Courier New
left_handed	<string>	Left handed switching of mouse and/or keyboard	NONE, MOUSE, KEYBOARD, ALL	NONE
zoom_factor	<real>	Zoom Factor for mouse wheel (0.01-1.0)	0.01 - 1.0	0.05
czoom_factor	<real>	Factor for right mouse dynamic zoom (0.01-0.2)	0.01 - 0.2	0.05
kzoom_factor	<real>	Factor for +/- keyboard short-cut keys	0.01 - 100.0	2.0
menu_dragging_mode	<string>	Mode used when moving menu panels with the mouse	WIREFRAME, OPAQUE	WIREFRAME
mouse_3d_rotation_factor	<real>	Factor applied to the speed of rotation when using a 3D mouse		1.0

mouse_3d_pan_factor	<real>	Factor applied to the speed of panning when using a 3D mouse		1.0
mouse_3d_zoom_factor	<real>	Factor applied to the speed of zooming when using a 3D mouse		1.0
mouse_action_middle_button	<string>	Set the action for the middle mouse key during picking	APPLY, REJECT, DESELECT	REJECT
mouse_action_right_button	<string>	Set the action for the right mouse key during picking	APPLY, REJECT, DESELECT	DESELECT

The following control treatment of recent files popups

Preference	**Type**	**Description**	**Valid arguments**	**Default**
recent_files_dropdown	<string>	Turn the recent files popup on or off	OFF, ON	ON
recent_files_max_but	<integer>	Maximum number of buttons displayed in a recent files popup	1 - 50	10
recent_files_max_char	<integer>	Maximum number of characters displayed on each recent files button	1 - 512	50

The following control treatment of unicode

Preference	**Type**	**Description**	**Valid arguments**	**Default**
cjk_unix_font	<string>	Font to use for CJK text on unix machines		-misc-fixed-medium-r-normal-*-12-*-*-*-*-*-*-*
cjk_windows_font	<string>	Font to use for CJK text on windows machines		MS Gothic 10
file_encoding	<string>	Character encoding for script files	Latin-1, BIG5, EUC-CN, EUC-JP, EUC-KR, GB, GBK, ISO-2022-CN, ISO-2022-CN-EXT, ISO-2022-JP, ISO-2022-JP-2, ISO-2022-KR, JOHAB, Shift-	Latin-1

			JIS, UTF-8, UTF-16BE, UTF-16LE, UTF-16, UTF-32BE, UTF-32LE, UTF-32	

The drive mappings allow PRIMER to convert equivalent folder names from Windows to Unix and visa versa. This is currently only in use for the JavaScript function DriveMapFilename for D3PLOT and T/HIS.

Preference	**Type**	**Description**	**Valid arguments**	**Default**
drive_a	<string>	Mapping from Windows drive A: to unix path		<none>
drive_b	<string>	Mapping from Windows drive B: to unix path		<none>
drive_c	<string>	Mapping from Windows drive C: to unix path		<none>
drive_d	<string>	Mapping from Windows drive D: to unix path		<none>
drive_e	<string>	Mapping from Windows drive E: to unix path		<none>
drive_f	<string>	Mapping from Windows drive F: to unix path		<none>
drive_g	<string>	Mapping from Windows drive G: to unix path		<none>
drive_h	<string>	Mapping from Windows drive H: to unix path		<none>
drive_i	<string>	Mapping from Windows drive I: to unix path		<none>
drive_j	<string>	Mapping from Windows drive J: to unix path		<none>
drive_k	<string>	Mapping from Windows drive K: to unix path		<none>
drive_l	<string>	Mapping from Windows drive L: to unix path		<none>
drive_m	<string>	Mapping from Windows drive M: to unix path		<none>
drive_n	<string>	Mapping from Windows drive N: to unix path		<none>
drive_o	<string>	Mapping from Windows drive O: to unix path		<none>
drive_p	<string>	Mapping from Windows drive P: to unix path		<none>
drive_q	<string>	Mapping from Windows drive Q: to unix path		<none>
drive_r	<string>	Mapping from Windows drive R: to unix path		<none>

drive_s	<string>	Mapping from Windows drive S: to unix path		<none>
drive_t	<string>	Mapping from Windows drive T: to unix path		<none>
drive_u	<string>	Mapping from Windows drive U: to unix path		<none>
drive_v	<string>	Mapping from Windows drive V: to unix path		<none>
drive_w	<string>	Mapping from Windows drive W: to unix path		<none>
drive_x	<string>	Mapping from Windows drive X: to unix path		<none>
drive_y	<string>	Mapping from Windows drive Y: to unix path		<none>
drive_z	<string>	Mapping from Windows drive Z: to unix path		<none>

The following control workflows functionality

Preference	**Type**	**Description**	**Valid arguments**	**Default**
workflow_definitions_directory	<string>	Location that will be scanned for Workflow definitions		<none>
workflow_only_use_specified_directory	<logical>	Only scan location set by preference oasys*workflow_definitions_directory for Workflow definitions (if it is set)	TRUE, FALSE	FALSE
workflow_user_data_directory_name	<string>	Name of a folder to search in for workflow user data		<none>
workflow_auto_open_post_menu	<logical>	Automatically open the Workflow menu in D3PLOT or T/HIS when reading in a model that has workflow data	TRUE, FALSE	FALSE
workflow_max_upward_folder_search_	<integer	Maximum number of folders to	0 - 100	4

20.3. C. Command - Windows File Associations

C. COMMAND - WINDOWS FILE ASSOCIATIONS

Under Windows on PC platforms it is possible to set up file associations so that double clicking on files with the `.ptf` , extension opens them automatically in D3PLOT.

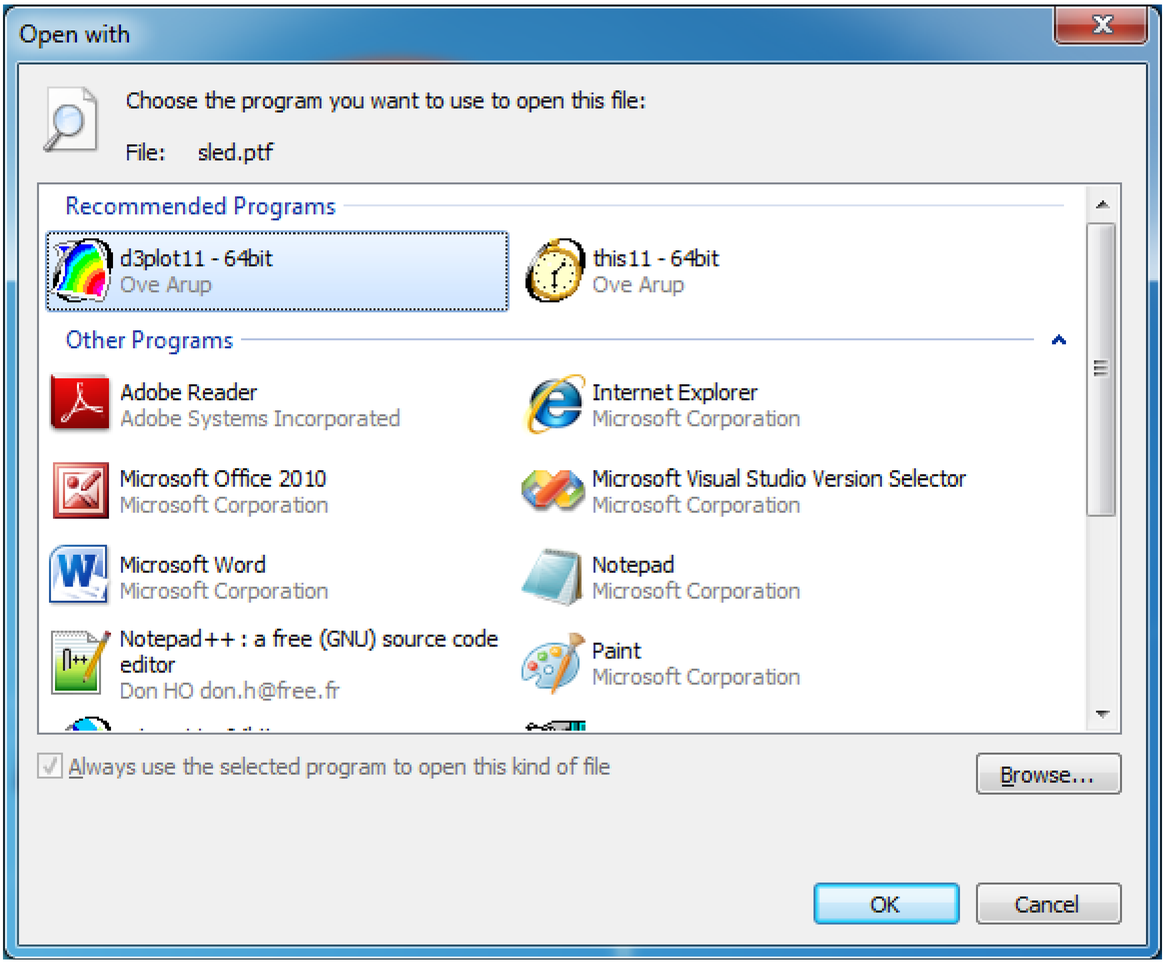
All of these settings are optional: you should be aware that under the Windows operating system associating a filetype (via its extension) with an application is convenient, but can also be restricting and complicated to undo.

20.3.1. To Make .ptf Files Open in D3PLOT by Double Clicking on Them

To make .ptf files open in D3PLOT by double-clicking on them

If no application is currently associated with .ptf files, a "double-click" won't work, and some non-specific, usually "windows", icon will be displayed with the file.

Right click on any **.ptf** file, and select **properties** then press the Change... tab next to Opens with: from the popup menu.



1. This will bring up the **"Open with"** panel.
2. Ensure the **Always use...** box is ticked
3. Use the directory browsing window to find the correct D3PLOT executable. You are looking for file **d3plot11.exe** or **d3plot11_x64.exe** .
4. Select the executable and click on **OK** to close the "Open With" window.

D3PLOT should now open and read in the selected file and you should now find that:

- All **.ptf** files on your system show the D3PLOT icon.
- Double-clicking on any such file starts D3PLOT and opens that file.

It is not possible to set up the filename "d3plot" for double-clicking in this way since Windows requires filename extensions when assigning applications to files.)

20.4. D. Environment Variables Used by D3PLOT

D. ENVIRONMENT VARIABLES USED BY D3PLOT

Environment variables can be used to set certain key parameters and to alter the default behaviour of the code. Generally these settings will be inserted into the Shell, but individual users are free to define their own parameters.

The setting of environment variables is done as follows:

20.4.1. Unix/Linux Systems Running "C" Shell (bin/csh) or its Derivatives such as /bin/tcsh

Unix/Linux systems running "C" shell (/bin/csh) or its derivatives such as /bin/tcsh

The format of the command is:

```
setenv <parameter> <argument list>
```

For example:

```
setenv DISPLAY my_machine:0
setenv SM_USE_VISUAL default
setenv DISPLAY_FACTOR 1.2
```

(Note that the Shell is written using C shell syntax, so if it is amended the format above should be used.)

20.4.2. Unix/Linux Systems Running "Bourne" (/bin/sh) or "Korn" (/bin/ksh) Shells

Unix/Linux systems running "Bourne" (/bin/sh) or "Korn" (/bin/ksh) shells

The format of the command is:

```
<parameter> = <argument list> ; export <parameter>
```

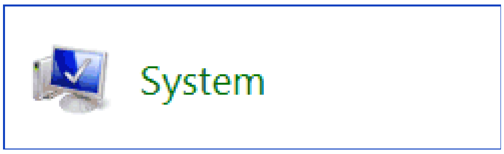
For example:

```
DISPLAY=my_machine:0; export DISPLAY
SM_USE_VISUAL=default; export SM_USE_VISUAL
DISPLAY_FACTOR=1.2; export DISPLAY_FACTOR
```

20.4.3. Windows Systems

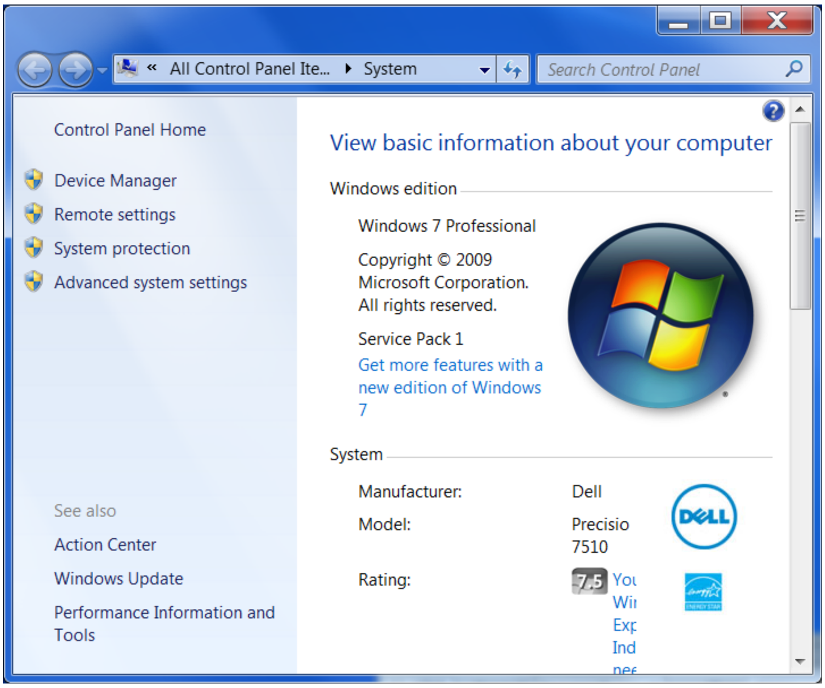
Windows systems

To set these system go into Control Panel and select the System Icon.

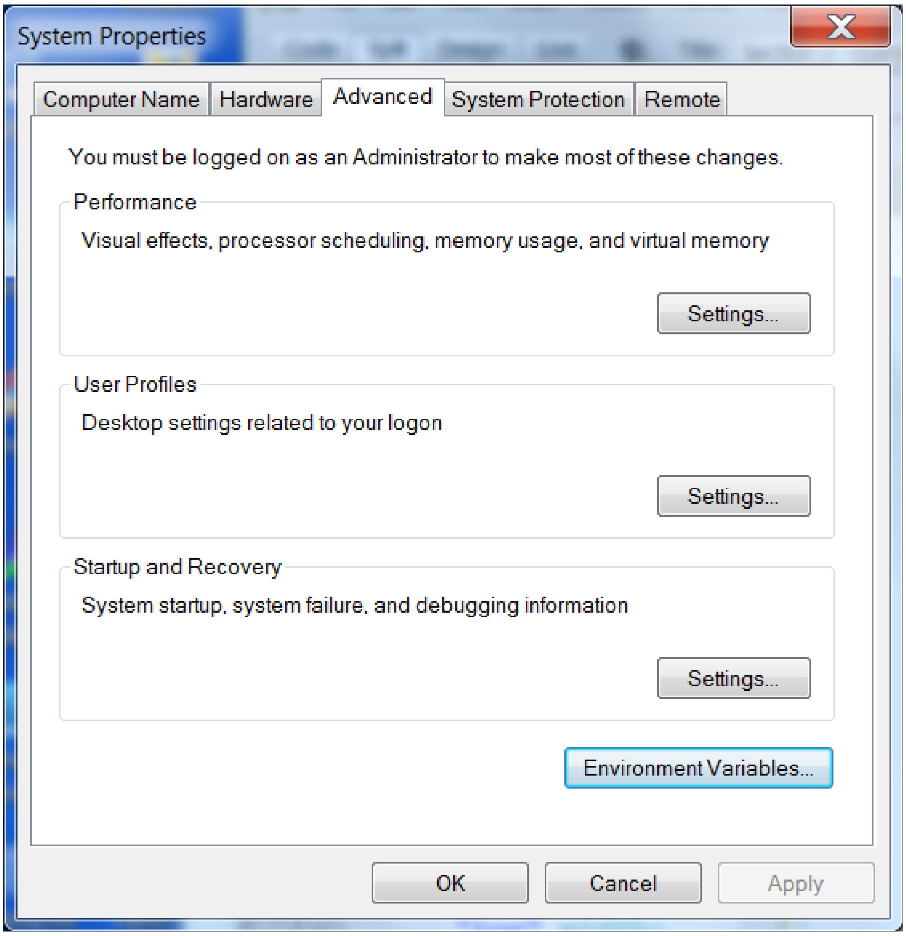


This will bring up the window displayed on the left.

Select the " **Advanced system settings** " option



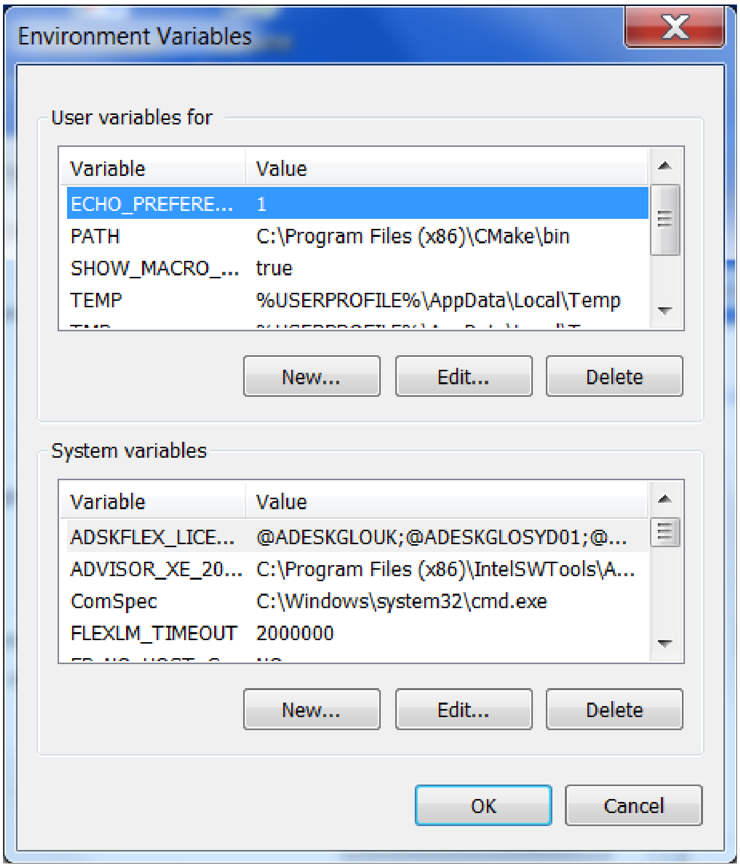
In the "System Properties" window that is displayed select the **Advanced** Tab and then the **Environment Variables** button



To set up a new Environment Variable select **New...**

Enter the variable's name in the Variable box followed by the variable's Value in the Value Box and then select **OK** .

Finally, after setting up all the new variables, select **Apply** followed by **OK** to dismiss the System Properties menu.



20.4.4. The Following Environment Variables May Be Used to Control the Behaviour of D3PLOT

The Following Enironment Variables May Be Used to Control the Behaviour of D3PLOT

Variable name	Description	Possible Values	Default
The following variables control the graphics and attributes of the display window and menu system.			
`DISPLAY`	The X11 display id on which graphics will be drawn. (This is ignored on "native" Windows systems.) If this is not defined (most systems initialise this to " `:0` ") then no connection can be made to an X	(<*machine name*>) `:<` ***`server id >`*** (<*.screen id*>)	`:0`

	server, and no graphics will be drawn.		
SM_USE_VISUAL	Sets the X11 "visual" id to be used for screen menus. Where a graphics display provides "overlay" planes these should normally be used, otherwise this should be left undefined or set to "default". Using an explicit visual id is possible, and this should be defined in hexadecimal (eg 0xf16). Experience has shown the on some Silicon Graphics systems using the "overlay" planes can result in very strange colours in other windows, in which case "default" should be used. Also on some W2000 and graphics board combinations problems may also arise with overlay planes and, again, "default" should be used.	**overlay** **default** < *visual id* > in hex	**overlay**
The following options may be used to configure the menu interface, and are intended for use by those who are having difficulties with the standard settings, or who have displays with unusual attributes. If you need advice about configuring your machine, and particularly if you suffer from physical impairment that may be eased by changing default settings, please contact Oasys Ltd and we will do our best to help.			
DISPLAY_SATURATION **DISPLAY_BRIGHTNESS** **DISPLAY_FACTOR**	Saturation controls the colour saturation (intensity) of menus Brightness controls the colour brightness of menus "Factor" sets the relative display scale, and can range from 0.5 (making menus larger) to 2.0 (making them smaller). It may also be set to " **automatic** " which	0.0 to 1.0 0.0 to 1.0 0.5 to 2.0, or **automatic**	**1.0** **1.0** **1.0**

	derives a factor from the physical screen dimensions.		
SM_FONT_SIZE	Sets the size of fonts used in the menu interface. Possible values are: **SMALL** Will reduce the size of text in windows and buttons, and may be suitable on very large displays. **MEDIUM** "Normal" text size suitable for most purposes. **LARGE** "Large" text, which may be more easily read by those with visual impairment. **DEFAULT** The default size (usually medium) on this hardware.	**SMALL** **MEDIUM** **LARGE** **DEFAULT**	< *none* > (implicitly **DEFAULT**)
SM_FONT_TYPE	Sets the font (type-face) to be used for menu fonts HELVETICA **HELVETICA-BOLD** Uses the proportionally spaced "Helvetica" (on X11) or "Arial" (on Windows) fonts. TIMES **TIMES-BOLD** Uses the proportionally spaced "Times" font. COURIER **COURIER-BOLD** Uses the fixed spaced "Courier" font.	**HELVETICA** **HELVETICA-BOLD** **TIMES** **TIMES-BOLD** **COURIER** **COURIER-BOLD**	**HELVETICA**

	The "Bold" options for the font give a bolder and thicker font. The actual type-face used on a given platform will depend upon what fonts have been installed on that machine, and it is possible that the font and size combination selected may not be available. For advice on installing fonts please contact Oasys Ltd.		
`LEFT_HANDED`	Configures the keyboard and/or mouse for left-handed users. **`KEYBOARD`** Swaps the left and right "meta" keys (<shift>and <ctrl>) for the purposes of dynamic viewing. ***Note*** *: At present (Oasys Suite 9.0) this does not work on Windows platforms for technical reasons. We hope to fix this in the future.* **`MOUSE`** Swaps the left and right mouse buttons for the purposes of all mouse actions (pick, drag, dynamic viewing) **`ALL`** Both of the above	**`KEYBOARD`** **`MOUSE`** **`ALL`**	*< none >*
The following two variables apply on **Windows platforms only** , and should only be used if the menu system is clearly obtaining the wrong display size from the system, as evidenced by fonts and menus being very much the wrong size.			

DISPLAY_HEIGHT	Set an explicit display height in millimetres	<height in mm>	<none>
DISPLAY_WIDTH	Set an explicit display width in millimetres	<width in mm>	<none>
The following options can be used to solve fairly obscure memory and display problems on X11 platforms.			
USE_PIXMAPS	Controls whether or not the menus use "pixmaps" (off-screen memory) to produce smooth scrolling. Turning this off (false) will save memory, and may help memory problems on a display that has only limited memory available for the X server, but will give slightly jerky window scrolling.	**true** or **false**	**true**
SAVE_UNDER	This flag was introduced to fix a specific bug on Compaq Alpha OSF4.x operating systems. Normally the window manager requests a redraw of windows that have been updated, even when they are currently obscured by something else. However the OSF4 window manager series failed to do this, leading to "bare" patches underneath popup menus when these were unmapped. Setting this flag to **false** results in more redraws on these systems since it suppresses the default "save under" property of X11 windows, but it does at least prevent windows getting bare areas. Compaq have fixed the bug in OSF5, and possibly in later releases of OSF4.	**true** or **false**	**true**

`ALPHA_PERMIT_BROKEN`	Another Compaq problem with older graphics cards under OSF4.x was a crash the first time broken lines for undeformed geometry were drawn. To prevent this Compaq machines have this capability switched off. To enable undeformed geometry to use broken lines on these machines set this variable to `true` .	`true` or `false`	`false`
The following are normally used when running command files, or performing automatic post-processing in batch mode.			
`MENU_AUTO_CONFIRM`	This variable is often used when replaying command files which, when recorded, paused and asked the user to confirm things. (For example HELP and Warning messages.) If the variable is set (true) then these will not pause and will behave as if the user had pressed "OK" - meaning that command files can play back without user intervention.	`true` or `false`	`false`

`FILE_EXIST_ACTION`	Action to be taken when opening a file for output, and the file already exists.
	
	Normally you will be prompted for the action to be taken when a file selected for output already exists. However if this variable is set to `overwrite` or `append` then the relevant action will be taken automatically.
	
	This is generally used when playing automatic post-processing batch scripts.
		< *none* >
		`overwrite`
		`append`
			< *none* >

<table>
<tr><td>SUPPRESS_CHECKPOINT</td><td>Suppresses the reading and writing of checkpoint files. This is desireable in batch mode since it prevents spurious checkpoint files being read in and processed.</td><td>true or false</td><td>false</td></tr>
<tr><td colspan="4">The following controls the display of on-line manual pages on Unix systems only. (Windows systems use the default web browser.)</td></tr>
<tr><td>NETSTART</td><td>Command string to start Netscape on Unix/Linux hosts. This is used to fire up the Netscape browser in order to read manual pages from within D3PLOT.</td><td>Any valid Unix command string.</td><td><none></td></tr>
<tr><td colspan="4">The following variables control the default behaviour of the database management system.</td></tr>
<tr><td>D3PLOT_CACHE_DATA</td><td>Default setting of the " CACHE_DATA " database switch. This controls whether or not D3PLOT attempts to store all data read from disk in its own core image. This setting can be changed manually during execution.
<table><tr><td>full (was true prior to V8.3)</td><td>Stores all data read from disk in memory until the database caching memory limits are reached. This can use a lot of memory and is not recommended unless disk access is very slow.</td></tr><tr><td>scalar</td><td>Stores basic nodal coordinates and the current component's "scalar" data only. A reasonable trade-off</td></tr></table></td><td>full
scalar
off

(Also, for backwards compatibility:

true
false</td><td>scalar</td></tr>
</table>

	<table><tr><td></td><td>between speed and memory size.</td></tr><tr><td>**off** (was **false** prior to v8.3)</td><td>Only basic nodal coordinate data are stored. Saves memory but may make data-bearing plots slow to modify.</td></tr></table> Generally Unix systems are better at cacheing disk data in spare system memory than Windows systems, making "disk" rereads faster since they are actually copied from memory. This makes the " **off** " option viable under Unix - however you may need to experiment.		
D3PLOT_SOFT_LIMIT	Controls the amout of memory (in MB) set aside for the "soft" database limit	1 to 2048 MB	60% of system memory
D3PLOT_HARD_LIMIT	Controls the amout of memory (in MB) set aside for the "hard" database limit	1 to 2048 MB	80% of system memory
The following variables set special parameters for data file reading and management, and are not normally used.			
D3PLOT_SOLID_SE	Controls whether or not D3PLOT calculates strain energy density for solid elements. Doing this makes the assumption that the elements have not entered the plastic strain regime, so it is not normally set.	**true** or **false**	**false**
D3PLOT_SOLID_8	Analyses containing 8 integration points worth of data for solid elements are processed (**NINTSLD** = 8 on ***DATABASE EXTENT BINARY**), but only the first integration point is normally considered. If this flag is set then limited support	**true** or **false**	**false**

	for all 8 integration points is provided. ***This facility is only partially implemented and should not be relied on.***		
`PTF_CONTIGUOUS` **`CTF_CONTIGUOUS`**	Some versions of LS-DYNA sometimes contravene the rules for starting a new family member, and write data contiguously across file boundaries when they should not. Setting these variables may enable such files to be read.	**`true`** or **`false`**	**`false`**
`D3PLOT_LIST_CROSSED`	Normally coincident solid elements and those with crossed faces are dealt with by issuing a summary warning that they exist. Set this variable to see a detailed listing of all such elements.	**`true`** or **`false`**	**`false`**
The following variables are provided for debugging purposes only, and should not normally be used.			
`XSYNC`	Runs the X server in "synchronised" (unbuffered) mode. This will give woefully slow graphics, and is used for debugging purposes only.	**`true`** or **`false`**	**`false`**
`WARN_REDEFINE`	Makes the menu system issue a warning if a button is redefined. Again this is normally only used for debugging purposes.	**`true`** or **`false`**	**`false`**
`CP_FILE_FILTER`	When replaying checkpoint files this maps the file filter box and waits for user input, instead of using the path/filename encoded in the checkpoint file. This is used when replaying checkpoint files on a machine different to that on which they	**`true`** or **`false`**	**`false`**

	were written, or when the encoded file pathnames are no longer valid. Though this will not work while replaying the checkpoint commands for the drag-drop of files into the D3PLOT graphics area.		
CP_DEBUG	When replaying checkpoint files this writes details of each command to <stdout>, and waits for <enter> before proceeding. **0** Off, the default. Checkpoint files play through without halts. **1** On. Checkpoint files echo all commands to <stdout>, and wait for a <return> on <stdin> before proceeding. However graphics commands (dynamic viewing, zoom, etc) play through without pause. **2** On. As for 1 above, but every command (including dynamic viewing) pauses and waits for confirmation before proceeding. Used for debugging problem files.	**0** (off) **1** **2**	**0**
CP_REFORMAT	When replaying checkpoint files written on a machine with a difference display resolution some picking and other screen-	**true** or **false**	**false**

	dependent operations may not work correctly. Setting this variable causes the playback machine to map the D3PLOT menu system onto a "virtual" display resolution equal to that of the original machine which *may* solve these problems if the two displays are not wildly different. It is more likely to work if the playback machine has a higher resolution than that on which the file was written.		
`D3PLOT_TIMING`	Writes the time taken for each	**`true`** or <	< *none* >

20.5. E. Dialogue Command Syntax

E. DIALOGUE COMMAND SYNTAX

D3PLOT has a dialogue command set that can be used in any of three ways:

1. In graphical (screen menu) mode commands can be typed into the Dialogue Box at any time.
2. In non-graphical (text only) mode commands are typed in at the terminal prompt
3. In command files, run either interactively or in batch, commands are executed as if typed in.

In all cases the command ***input*** syntax is identical, although there are minor differences in output between "screen menu" and "text-only" modes: in the latter case all output has to go to the controlling terminal ("stdout"), whereas in the former separate windows are used for "help", "listing" and other output.

For D3PLOT 20.0 onwards, we have now also added the ability to use the Up and Down arrow keys in order to cycle through previously issued commands that you have entered into the Dialogue window. Currently only the last 50 commands are stored in memory. Commands issued after this 50 limit will remove the oldest issued commands first from the list.

20.5.1. The Dialogue Command Structure

The Dialogue Command Structure

The command structure forms a hierarchical "tree", with the top-level **D3PLOT_MANAGER** at its "root".

The following rules apply:

- Command words may be abbreviated to any degree so long as:
  - they are unique in the context of their current menu
  - they must have at least their first two characters given

  For example **`BP_BEAM_PLOTTING CT_CONTINUOUS`** may be abbreviated to **`BP CT`** .

- Navigation up and down menu levels is performed as follows:
  - <command> takes you to the command's (sub-)menu level
  - Forward slash "/" takes you back to the top **D3PLOT_MANAGER** level before executing the following command(s)

  For example **`BP_BEAM_PLOTTING`** above takes you into the **BEAM_PLOTTING** sub-menu

The command **/DEFORM EXPLODE** would work at the **BEAM_PLOTTING** prompt because it would return to the top level before parsing the **DEFORM** command.

- There is also a "global menu" of commands which is available at any (sub-)menu prompt.
  - These are primarily graphics commands that do not require a context
  - The commands can be listed with the **GM** (for Global Menu) command

- Any command can be aborted by typing **Q** (uit). This will return control to the next highest command prompt in the "tree"
- At any prompt you can type **H** (elp) to receive advice about what to do next.
- For all the places where <entity><type> range needs to be specified it can be given as

  A range of numbers using typical syntax:
  1) 1 2 3 19 207 5 8
  2) 1 TO 10 3 6 20 TO 100 STEP 1
  3) ALL (or *)
  4) AV (or %) for all visible entities

  Using the screen picking options:
  1) V (Visible) To pick individually with cursor;
  2) SA (Screen Area) to pick entities within a box defined by opposite corners;
  3) CV (Current Volume) to pick entities inside the current clipping volume.

20.5.2. Main Menu Commands

CT_CONTINUOUS_TONE	Continuous tone (solid) contour plot
LC_LINE_CONTOURS	Line contour plot
CL_CLOUD_PLOT	"Cloud" (points) plot
ISO_SURFACE_PLOT	Iso-surface contour plot
VELOCITY_PLOT	Velocity arrows plot
CRITERION_PLOT	Criterion mode plot

The syntax is:

COMPONENT <data>	Defines the data component <data> to plot
ATTRIBUTE <attr>	Defines the attributes of the criterion analysis. Valid <attr>s are:

LENGTH <length>	Symbol length
COLOUR <colour>	Symbol colour. Valid <colour>s are: **DATA_VALUE** Scale colours with data values **FIXED_COLOURS** Sets fixed colours
SYMBOL <symbol>	Symbol type. Valid <symbol>s are: **HIERARCHY** MAX stress has normal arrow head MID stress has flat T arrowhead MIN stress has inverted arrow head **ALL_LINES** All components drawn using plain lines

GO	Perform the criterion plot
EXPLAIN	Further help
STATUS	Show the current settings

CT_CONTINUOUS_TONE	Continuous tone (solid) contour plot
SI_SHADED_IMAGE	Shaded image plotting options The syntax is:

SHININESS <%age>	Set the object shininess
SATURATION <%age>	Set the object colour saturation
OVERLAY_COLOUR <colour>	Set the overlay colour
LM_LIGHTING_MODEL <switch>	Turn lighting model ON or OFF
BRIGHTNESS <%age>	Set directional lighting intensity
AMBIENT_LIGHT_LEVEL <%age>	Set ambient light level
SHADING_TYPE <option>	Set smooth or flat shading. Valid <option>s are:

FLAT	Flat shading
SMOOTH <angle>	Smooth shading with an edge angle <angle>

DS_DITHER_SHADING <switch>	Turn dithered shades ON or OFF
GOURAD_SHADING <option>	Set solid or fuzzy contour bands. Valid <option>s are:

ON	Fuzzy bands
OFF	Solid bands

DC_DITHER_CONTOURS <switch>	Turn contour dithering ON or OFF
COLOUR_MAP	Draw the current colour map
GO	Execute the plot
STATUS	Show the current settings
EXPLAIN	Further help

INTERFACE_PLOTTING

Plot sliding interface results

The syntax is:

COMPONENT <data>	Define the <data> component to plot
LC_LINE_CONTOURS	Line plot

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

CT_CONTINUOUS_TONE	Continuous tone contour plot
SI_SHADED_IMAGE	Shaded contour plot
VECTOR_PLOT	Vector force plot
HATCHING_SWITCH <switch>	<switch> hatching ON or OFF
OPACITY_SWITCH <switch>	<switch> opacity ON or OFF
EXPLAIN	Further help

BP_BEAM_PLOTTING

Plots beam data

The syntax is:

CT_CONTINUOUS	Continuous tone contour plot
DP_DIAGRAM_PLOT	Bending moment diagram plot
COMPONENT <data>	Defines the data component <data> to plot
INTG_POINT	Set H-L beam extra dat integration point
R2_REVERSE_END_2	Set the reverse end 2 switch on or off
THICKNESS	Set the CT plot beam thickness
ATTRIBUTES <attr>	Defines the attributes of the DP plot. Valid <attr>s are:

SIZE <size>	Size of the maximum diagram vector in screen units
HATCHING <hatch>	Intervals between the intermediate lines in screen units
PROJECTION <proj>	How the diagram is projected. Valid <proj>s are: **SCREEN** Always in the screen XY plane **LOCAL** In the local YY or ZZ plane
LABEL_VALUES	Set the label switch on or off

OPACITY_SWITCH	Set the opacity switch on or off
EXPLAIN	Further help
WARNING	Some notes and caveats

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

STATUS	Show the current settings

OTHER_PLOTTING

Plots other (LSDA) data

The syntax is:

CT_CONTINUOUS	Continuous tone contour plot
SI_SHADED_IMAGE	Shaded contour plot
COMPONENT <data>	Defines the data component <data> to plot

GREYSCALE

Draw a solid shaded plot

The syntax is:

SHININESS <%age>	Set object shininess level
SATURATION <%age>	Set object colour saturation
OVERLAY_COLOUR <colour>	Set overlay colour
LM_LIGHTING_MODEL <switch>	Turn lighting model ON or OFF
BRIGHTNESS <%age>	Set directional light intensity
AMBIENT_LIGHT_LEVEL <%age>	Set ambient light level
SHADING_TYPE <option>	Set smooth or flat shading. Valid <option>s are:

FLAT	Flat shading
SMOOTH <angle>	Smooth shading with an edge angle <angle>

DS_DITHER_SHADING <switch>	Turn dithered shades ON or OFF
GO	Execute the plot
STATUS	Show the current settings
EXPLAIN	Further help

ENVELOPE_PLOT

COMPONENT <type>	Select the type of value to plot. Valid <type>s are:

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

OFF	Turn envelope plotting off
MAX_VALUE	Plot max values
TIME_OF_MAX_VALUE	Plot time of the max values
MIN_VALUE	Plot min values
TIME_OF_MIN_VALUE	Plot time of min values
ABS_VALUE	Plot absolute value
TIME_OF_ABS_VALUE	Plot time of absolute values

STATES <list> Give a <list> of state numbers to be used

MAX_MIN

Controls max / min value plotting

The syntax is:

OFF	Turn off display of max/min values
ON_DATA_PLOTS	Turn on display of max/min values on data bearing plots
ON_ALL_PLOTS	Turn on display of max/min values on all plots
NUMBER_OF_VALUES	Set the number of values shown
(NO_)MAX_LIST	Turn on/off max values in a list
(NO_)MAX_LABEL	Turn on/off max labels on plot
(NO_)MAX_VALUES	Turn on/off max values on plot
(NO_)MIN_LIST	Turn on/off min values in a list
(NO_)MIN_LABEL	Turn on/off min labels on plot
(NO_)MIN_VALUES	Turn on/off min values on plot

COMPONENT <data> Defines the data component <data> to plot

STRESS_CONTROL

Control stress plotting

The syntax is:

PART_IGNORED_SW	Do / do not average across parts
BLANKING_IGNORED_SW	Do / do not include blanked elements
CLIPPING_IGNORED_SW	Do / do not include clipped elements

CT_CONTINUOUS_TONE	Continuous tone (solid) contour plot	
	AVERAGE_SWITCH	Do / do not average stress at nodes
	GLOBAL_COORDINATES	Use global coordinates
	LOCAL_COORDINATES	Rotate to local element coordinates
	CYLINDRICAL_COORDINATES	Rotate to cylindrical coordinates
	USER_DEFINED_COORDINATES	Rotate to user defined coordinates
	PLY_LOCAL	Rotate to ply local system
	MATERIAL	Rotate to material axes coordinate system
	SURFACE <surface>	Select a shell surface or layer. Valid <surface>s are: TOP, MIDDLE, BOTTOM, MAX, MIN, MAG or a layer number if present
	PLY_SELECT <list>	Select a <list> of ply(s) to plot data on
	EXPLAIN	Further help
	STATUS	Show the current settings
CONTOUR	Control contour settings The syntax is:	
	NUMBER_OF_LEVELS <levels>	Defines the number of contour levels
	AUTOMATIC	Autoscales the levels
	MANUAL_MAX_MIN <min> <max>	Manually sets the minimum <min> and maximum <max>
	USER_DEFINED	Define each contour line
	REVERSE	Reverse the contour colours
	LINEAR_SCALE	Convert to linear scale

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

LOG_SCALE	Convert to logarithmic scale
RESOLUTION <res>	Define the contour resolution. Valid <res>s are: LOW, MEDIUM or HIGH
FORCE_LABELS	Force line labelling on LC plots
LABEL_FREQUENCY <freq>	Set the labelling frequency on LC plots to <freq>
LIMITING_VALUES <switch> <low> <high> ***(on new line)*** **<action>**	Define what is contoured by upper and lower bound limits. Where: <switch> Is either ON or OFF <low> Is the lower limit <high> Is the upper limit <action> Is the exclusion behaviour and can be: OMIT, OUTLINE or BLACK
COLOURS <band> ***(on new line)*** **<colour>**	Set the colour of an individual contour band
OVERLAY_COLOUR <colour>	Set the overlay colour
ARROW_LENGTH <length>	Set the length of arrows
FORMAT_NUMBER <option>	Set the format of the numbers on the contour bar. Valid <option>s are:

AUTOMATIC	D3PLOT will set it automatically
SCIENTIFIC	Scientific format
GENERAL	General format
MANUAL	Manual format
EXPONENT_VALUE <value>	Set the exponent value
DECIMAL_PLACES <value>	Set the number of decimal places

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

TSHELL_OPTS <opt>	Set the thick shell contour method. Valid <opt>s are: INTERPOLATED or SIMPLE
STATUS	Show the current settings

DISPLAY_OPTIONS

Set display parameters

The syntax is:

UNDEFORMED_SWITCH	Do / do not draw undeformed geometry
BF_SWITCH	Do / do not draw back faces
IF_SWITCH	Do / do not draw internal faces
LOCAL_TRIAD_SWITCH	Do / do not draw triad
CLOCK_SWITCH	Do / do not draw clock
HEADER_SWITCH	Do / do not draw title header
DISP_MAG_SWITCH	Do / do not draw displacement magnifications
CONTOUR_BAR_SWITCH	Do / do not draw contour bar
NASTRAN_CASES_SW	Do / do not draw nastran cases/freqs
DATE_SWITCH	Do / do not draw date on plots
BORDER_SWITCH	Do / do not draw border
GRATICULE_SWITCH	Do / do not draw graticule
GRID_SWITCH	Do / do not draw graticule grid
G3D_GRATICULE	Turn on / off the 3D graticule
X_GRATICULE	Turn on / off the 3D X graticule plane
Y_GRATICULE	Turn on / off the 3D Y graticule plane

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

Z_GRATICULE	Turn on / off the 3D Z graticule plane
X_GRAT_POS <pos>	Set the position of the X graticule to <pos>
Y_GRAT_POS <pos>	Set the position of the Y graticule to <pos>
Z_GRAT_POS <pos>	Set the position of the Z graticule to <pos>
GRAT_NUMBER_FORMAT <option>	Set the number format on the graticule. Valid <option>s are:

AUTOMATIC	D3LOT will set it automatically
SCIENTIFIC	Scientific format
GENERAL	General format
MANUAL	Manual format
EXPONENT <value>	Set the exponent value
DECIMAL_PLACES <value>	Set the number of decimal places

GRAT_PLANE_COLOUR <col>	Set the colour of the graticule plane to <col>
GRAT_LINE_COLOUR <col>	Set the colour of the graticule lines to <col>
GRAT_TEXT_COLOUR <col>	Set the colour of the graticule text to <col>
MODEL_BOX_SWITCH	Do / do not draw box round model
ALL_NODES_SWITCH	Do / do not draw all nodes
LABEL_SWITCH	Control node / element labelling
ENTITY_SWITCH	Control entity visibility
HO_HIDDEN_LINE_OPT <option>	Set hidden line options. Valid <option>s are:

PAINTER	Use "painter" algorithm
RIGOROUS	Use "rigorous" algorithm
RESOLUTION <x> <y>	Set the resolution to <x> x <y>

FREE_FACE_OPTIONS <option>	Set how free edges are displayed. Valid <option>s are:

OFF	Turn free edges off

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

ON	Turn free edges on
BE_BLANK_EDGES	Blanking does create free edges
BN_BLANK_NO_EDGES	Blanking does not create free edges
CE_CLIP_EDGES	Clipping does create free edges
CN_CLIP_NO_EDGES	Clipping does not create free edges
PE_PART_EDGES	Part boundaries create edges
PN_PART_NO_EDGES	Part boundaries do not create edges
SE_SURF_EDGES	Surface boundaries create edges
SN_SURF_NO_EDGES	Surface boundaries do not create edges

SEAT_BELT_OPTIONS <option> Set how seat belt elements are displayed. Valid <option>s are:

BELT_WIDTH <width>	Set the visual width
RETRACTOR_SIZE <size>	Set the visual retractor size
SLIP_RING_SIZE <size>	Set the visual slip ring size

SPRING_SYMBOL <option> Set how spring elements are displayed. Valid <option>s are:

ZIG_ZAG	Display springs as a zig-zag
LINE	Display springs as a line

BEAM_SYMBOL <option> Set how beam elements are displayed. Valid <option>s are:

LINE	Display beams as a thin line
THICK_LINE	Display beams as a thick line

WINDOW <option> Define the image window setting. Valid <option>s are:

FULL_SCREEN	The plot occupies the full screen
PART_SCREEN	The plot will not overwrite title or contour key

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

REPORT_FORMAT	The colour hard-copied the image will be the same size as a Laser A4 plot
USER_DEFINED	Define the plot window with cursor

STATUS Show the current settings

PROPERTIES

Set model properties

The syntax is:

SAVE <mod> **(on new line) <filename>**	Save a properties file for model <mod> to <filename>
LOAD <mod> **(on new line) <filename>**	Load a properties file for model <mod> from <filename>
DISPLAY_MODE <entity> <list> ***(on new line)* <mode>**	Set the display mode of a <list> of <entity> types to <mode>. Valid <mode>s are:

WIRE	Wireframe mode
HIDDEN	Hidden Line mode
SHADED	Shaded mode
CURRENT	Current mode

COLOUR <entity> <list> **(on new line) <col>**	Set the colour of a <list> of <entity> type to <col>
TRANSPARENCY <entity> <list> **(on new line) <trans>**	Set the transparency of a <list> of <entity> type to <trans> %
BRIGHTNESS <entity> <list> **(on new line) <bright>**	Set the brightness of a <list> of <entity> type to <bright> %
SHININESS <entity> <list> ***(on new line)* <shine>**	Set the shininess of a <list> of <entity> type to <shine> %
OC_OVERLAY_COLOUR <entity> <list> **(on new line) <col>**	Set the overlay colour of a <list> of <entity> type to <col>
OM_OVERLAY_MODE <entity> <list> **(on new line) <mode>**	Set the overlay mode of a <list> of <entity> types to <mode>. Valid <mode>s are:

NONE	No overlay
FREE_EDGES	Overlay on free edges

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

FULL	All overlay
CURRENT	Current overlay

More info on valid <list> arguments can be found here

VIEW_OPTIONS

Store and get views from a file

The syntax is:

STORE <view id> **(on new line) <name>**	Store the current view in <view id> with the name <name>
GET <view id>	Get the view <view id>
DIRECTORY	List the stored views
RENAME <view id> ***(on new line)* <name>**	Rename a stored view <view id> to <name>
DELETE <view id>	Delete view <view id>
FILE_NAME <file_name>	Change the name of the file the views are stored in to <file_name>
PERSPECTIVE <option>	Set perspective options. Valid <option>s are:

ON	Turn perspective on
OFF	Turn perspective off
DISTANCE <distance>	Set the eye to centre distance to <distance>

WE_WRITE_EXPLICIT	Display the explicit centre and scale
RE_READ_EXPLICIT <x, y, z> <scale>	Set the explicit centre to the coordinates <x y z> and the scale to <scale>
STATUS	Show the current settings
EXPLAIN	Further help

CT_CONTINUOUS_TONE	Continuous tone (solid) contour plot

BLANK

Blank / unblank entities

The syntax is:

<entity> <list>	Blank the <list> of entities of type <entity>
ON	Switch blanking on
OFF	Switch blanking off
ALL	Blank all entities except nodes
UNBLANK <entity> <list>	Unblank the list of entities of type <entity>
REVERSE	Reverse the blanking
STATUS	Show the current setting
	More info on valid <list> arguments can be found here

UNBLANK

Unblank entities

The syntax is:

<entity> <list>	Unblank the <list> of entities of type <entity>
ALL	Unblank all entities
	More info on valid <list> arguments can be found here

VOLUME_CLIPPING

Clip display by volume

The syntax is:

CREATE <option>	Create a clipping volume. Valid <option>s are:

CARTESIAN <x_c, y_c, z_c> (on new line) <x, y, z> (on new line) <orient>	Define a box. Where: <x_c, y_c z_c> — Defines the centre of the box <x, y z> — Defines the dimensions of the box

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

	<orient>	Defines the orientation of the volume and can be: BASIC_MODEL, DEFORMED_MODEL or SCREEN_SPACE

NODE_CARTESIAN <nid> ***(on new line)* <x, y, z>** ***(on new line)* <orient>**	Define a box. Where:	
	<nid>	Is a node that defines the centre of the box
	<x, y z>	Defines the dimensions of the box
	<orient>	Defines the orientation of the volume and can be: BASIC_MODEL, DEFORMED_MODEL or SCREEN_SPACE

CYLINDRICAL <x_c, y_c, z_c> ***(on new line)* <rad>** **<min_height> <max_height>** ***(on new line)* <H_axis>** ***(on new line)* <orient>**	Define a cylinder Where:	
	<x_c, y_c, z_c>	Defines the centre of the cylinder
	<rad>	Defines the radius of the cylinder
	<min_height> <max_height>	Define the height of the cylinder
	<H_axis>	Is the global axis to align the cylinder with and can be X, Y or Z

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

	<orient> — Defines the orientation of the volume and can be: BASIC_MODEL, DEFORMED_MODEL or SCREEN_SPACE

NODE_CYLINDRICAL <nid> ***(on new line)*** **<rad> <min_height> <max_height>** ***(on new line)*** **<H_axis>** ***(on new line)*** **<orient>**	Define a cylinder Where: <nid> — Is a node that defines the centre of the cylinder <rad> — Defines the radius of the cylinder <min_height> <max_height> — Define the height of the cylinder <H_axis> — Is the global axis to align the cylinder with and can be X, Y or Z <orient> — Defines the orientation of the volume and can be: BASIC_MODEL, DEFORMED_MODEL or SCREEN_SPACE

SPHERICAL <x_c, y_c, z_c> ***(on new line)*** **<rad>** ***(on new line)*** **<orient>**	Define a sphere Where: <x_c, y_c, z_c> — Defines the centre of the sphere <rad> — Defines the radius of the sphere

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

	<orient> Defines the orientation of the volume and can be: BASIC_MODEL, DEFORMED_MODEL or SCREEN_SPACE

NODE_SPHERICAL <nid> ***(on new line)*** **<rad>** ***(on new line)*** **<orient>**	Define a sphere Where: <nid> Is a node that defines the centre of the sphere <rad> Defines the radius of the sphere <orient> Defines the orientation of the volume and can be: BASIC_MODEL, DEFORMED_MODEL or SCREEN_SPACE

CANCEL	Cancels the clipping volume
ORIENT	Defines the orientation of the volume and can be: BASIC_MODEL, DEFORMED_MODEL or SCREEN_SPACE
DO_DISCARD_OUTSIDE	Remove elements outside the volume
DI_DISCARD_INSIDE	Remove elements inside the volume
OFF	Turn the clipping off

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

ON	Turn the clipping on
DRAW_OFF	Turns drawing of volume off
DRAW_ON	Turns drawing of volume on
FNODE_OFF	Turns the follow node switch off
FNODE_ON	Turns the follow node switch on
LOCATION_PLOT	Draw 4 views of the model to visualise the volume
STORE <volume id> ***(on new line)*** **<name>**	Store the current volume in <volume id> with the name <name>
GET <volume id>	Retrieve the stored volume <volume id>
RENAME <volume id> ***(on new line)*** **<name>**	Rename a stored volume <volume id> to <name>
DELETE <volume id>	Delete volume <volume id>
DIRECTORY	List stored volumes on file
FILE_NAME <file_name>	Change the name the volumes are stored in to <file_name>
STATUS	Show the current settings
EXPLAIN	Further help

CUT_SECTIONS

Define a section that cuts through the model. Note that currently only the first non parallel direction is supported. For automation using these options for multiple directions please use the JavaScript API.

The syntax is:

OFF	Turn the cut section off
ON	Turn the cut section on
CREATE <option>	Create a cut section. Valid <option>s are:

LS_DYNA_METHOD **<x_nt, y_nt, z_nt>** ***(on new line)*** **<x_nh, y_nh, z_nh>** ***(on new line)*** **<x_eh, y_eh,**	Define normal and edge coordinates

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

z_eh> ***(on new line)*** **<orient>**	Where: <x_nt, y_nt, z_nt> — Defines the Normal Tail coordinate <x_nh, y_nh, z_nh> — Defines the Normal Head coordinate <x_eh, y_eh, z_eh> — Defines the Edge Head coordinate <orient> — Orientation of cutting plane and can be: BASIC_MODEL, DEFORMED_MODEL, SCREEN_SPACE

OV_ORIGIN_&_VECTORS <x_o, y_o, z_o> ***(on new line)*** **<x_x, y_x, z_x>** ***(on new line)*** **<x_xy, y_xy, z_xy>** ***(on new line)*** **<orient>**	Define an origin, local x axis and local xy plane Where: <x_o, y_o, z_o> — Defines the origin <x_x, y_x, z_x> — Defines the local x axis <x_xy, y_xy, z_xy> — Defines the local xy plane <orient> — Orientation of cutting plane and can be: BASIC_MODEL, DEFORMED_MODEL, SCREEN_SPACE

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

N3_THREE_NODES <N1, N2, N3> ***(on new line)*** **<orient>**	Define a plane with three nodes Where: <N1, N2, N3> — Defines the three nodes <orient> — Orientation of cutting plane and can be: BASIC_MODEL, DEFORMED_MODEL, SCREEN_SPACE

X_CONSTANT <x> ***(on new line)*** **<orient>**	Define a plane of constant X Where: <x> — Defines the x coordinate <orient> — Orientation of cutting plane and can be: BASIC_MODEL, DEFORMED_MODEL, SCREEN_SPACE

Y_CONSTANT <y> ***(on new line)*** **<orient>**	Define a plane of constant Y Where: <y> — Defines the y coordinate <orient> — Orientation of cutting plane and can be:

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

	BASIC_MODEL, DEFORMED_MODEL, SCREEN_SPACE

Z_CONSTANT <z> ***(on new line)*** **<orient>**	Define a plane of constant Z Where: <z> Defines the z coordinate <orient> Orientation of cutting plane and can be: BASIC_MODEL, DEFORMED_MODEL, SCREEN_SPACE

CANCEL Cancels the cut section

SYSTEM <orient> Define the orientation of the cutting plane. Valid <orient>s are:

BASIC_MODEL	Tied to model space, undeformed geometry
DEFORMED_MODEL	Tied to model space, deformed geometry
SCREEN_SPACE	Tied to the screen

FSYS <system>

Define the system for force and moment calculations. Valid <system>s are:

AUTOMATIC	Basic space will use global coordinates, deformed and screen space will use local coordinates
GLOBAL	Local coordinates

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

LOCAL	Global coordinates

POSITIVE_ACTION <action>	Action for elements on positive side of the section. Valid <action>s are:

OMIT	Do not draw the elements
OUTLINE	Draw the elements in outline
NORMAL	Draw the elements in the display mode

NEGATIVE_ACTION <action>	Action for elements on negative side of the section. Valid <action>s are the same as above

THICK_CUT

	<thickness>	Set the plane thickness, eg: **thick_cut 100.0** **thick_cut off**
or	**OFF**	

CAPPING <action>	Action for capping 2D elements. Valid <action>s are:

OFF	Do not cap 2D elements
TRUE_THICKNESS <factor>	Cap 2D elements with their true thickness * <factor>
FIXED_THICKNESS <value>	Cap 2D elements with a fixed thickness of <value>

LOCATION_PLOT	Draw 4 views of the model to visualise the section
VIEW_PLANE	Change view to normal to plane
SKETCH	Sketch the section
FORCE	Calculate forces on current section
WRITE_FORCES <name>	Write forces to csv file <name>
STORE <section id> ***(on new line)*** **<name>**	Store the current section in <section id> with the name <name>

CT_CONTINUOUS_TONE	Continuous tone (solid) contour plot	
	GET <section id>	Retrieve the stored section <section id>
	RENAME <section id> ***(on new line)*** **<name>**	Rename a stored section <section id> to <name>
	DELETE <section id>	Delete section <section id>
	DIRECTORY	List stored sections on file
	FILE_NAME <file_name>	Change the name the sections are stored in to <file_name>
	STATUS	Show the current settings
	EXPLAIN	Further help

DEFORM

Modify or deform the geometry

The syntax is:

EXPLODE <action> — Separate parts. Valid <action>s are:

DEFINE <parts> ***(on new line)*** **<x, y, z>**	Define the list of <parts> to explode along the vector <x, y, z>
CANCEL	Remove the vector

MAGNIFY_DISP <x, y, z> — Magnify the displacements by <x, y, z>

FIX_NODE <action> — Apply a negative translation, equal to the displacement of a node, to the whole model. Valid <action>s are:

DEFINE <node id>	Define the node <node id>
CANCEL	Remove the translation

SHIFT_DEFORMED <action> — Apply a negative translation and rotation equal that defined by three nodes, to the whole model. Valid <action>s are:

DEFINE <N1, N2, N3>	Define the three nodes
CANCEL	Remove the translation and rotation

REF_NODE <action> — Contour relative to a node or three nodes. Valid <action>s are:

SINGLE_NODE <node id>	Define the node <node id>

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

THREE_NODES <N1, N2, N3>	Define the three nodes
GLOBAL	Contour results in global system
LOCAL	Contour results in local system defined by nodes<N1, N2, N3>
REF_VALUES	Toggle on and off whether reference values should be used for WRITE and XY_DATA
SHOW_NODES_ON	Show the selected node(s) in the graphics window
SHOW_NODES_OFF	Don't show the selected node(s) in the graphics window

REF_STATE <action> Contour relative to a state. Valid <action>s are:

ON	Turn reference state on
OFF	Turn reference state off
REF_MODEL <action>	Select a reference model. Valid <action>s are: **NUMBER <model id>** — Set the reference model to <model id> **CURRENT** — Set the reference model to the current model
SET_REF_STATE <action>	Select a reference state. Valid <action>s are: **NUMBER <state id>** — Set the reference state to <state id> **TIME_RS <time>** — Set the reference state to the state at time <time> **CURRENT** — Use the current state

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

COORDS	Toggle on or off whether to apply reference to current coordinates
DATA_VALUES	Toggle on or off whether to apply reference to data values
UNDEFORMED	Toggle on or off whether to apply reference to undeformed geometry

TRANSFORM <action>

TRANSLATE		**Tx Ty Tz**	Translate by Tx Ty Tz, eg: **translate 10.0 0.0 -100.0** **translate off**
	or	**OFF**	
REFLECT		**Axis Distance**	"Axis" is **X** or **Y** or **Z**, "distance" is position on axis, eg: **reflect Y -1500.0** **reflect off**
	or	**OFF**	
ROTATE		**Tx Ty Tz Cx Cy Cz**	Tx Ty Tz are rotation angles in degrees, Cx Cy Cz is centre of rotation, eg: **rotate 0 0 30 100.0 10.0 -20.0** **rotate off**
	or	**OFF**	
SCALE		**Sx Sy Sz**	Scaling by factors Sx Sy Sz, eg: **scale 2.0 2.0 2.0** **scale off**
	or	**OFF**	
CANCEL		<No arguments>	Turns off ***ALL*** transformations (leaving values unchanged)
STATUS		<No arguments>	Shows current transformation status

STATUS Show the current settings

EXPLAIN Further help

ATTACHED

Find attached items

The syntax is:

APPLY Find attached

SAVE_CURRENT Save current blanking status

RESTORE_SAVED Restore blanking status

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

RECURSIVE Set recursive attached options. Valid <option>s are:

ON	Turn on recursive find
OFF	Turn off recursive find
MAX_LOOPS <n>	Set the max number of loops to <n>

SELECTION <option> Valid <option>s are:

ATTACHED_PART	Find whole parts
SINGLE_ELEM	Find single elements

METHOD <option> Valid <option>s are:

VISIBLE	Find attached to what is visible
SELECTED <list>	Select <list> of nodes to find attached to

More info on valid <list> arguments can be found here

THROUGH <option> Toggle on and off the entities that find attached will look for. Valid <option>s are:

NODES ON/OFF	Turn on/off finding attached through nodes
SOLIDS ON/OFF	Turn on/off finding attached through solids
BEAMS ON/OFF	Turn on/off finding attached through beams
SHELLS ON/OFF	Turn on/off finding attached through shells
THICK_SHELLS ON/OFF	Turn on/off finding attached through thick shells

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

SPRINGS ON/OFF	Turn on/off finding attached through springs
CONNECTION_TYPES ON/OFF	Turn on/off finding attached through connection types

LAYOUT

Set the page layout

The syntax is:

SAME_SIZE <action>	Set switch to make tiled windows the same size. Valid <action>s are:

ON	Turn on switch to make tiled windows the same size
OFF	Turn off switch to make tiled windows the same size

WIDE	Set the page layout to be tile wide.
TALL	Set the page layout to be tile tall.
CASCADE	Set the page layout to be cascade.
ONE_X_ONE	Set the page layout to be 1x1.
TWO_X_TWO	Set the page layout to be 2x2.
THREE_X_THREE	Set the page layout to be 3x3.
XY <X> ***(on new line)* <Y>**	Set the page layout to be <X>x<Y>
CUSTOM <options>	Customise the page layout. Valid <option>s are:

LAYOUT <page> ***(on new line)* <layout>**	Set the layout of page <page> to <layout>. Valid <layout>s are: **WIDE** Tile wide **TALL** Tile tall **CASCADE** Cascade **ONE_X_ONE** 1x1 **TWO_X_TWO** 2x2 **THREE_X_THREE** 3x3 **XY <X> <Y>** <X>x<Y>

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

AW_ADD_WINDOW <wdw> *(on new line)* **<page>**	Add D3PLOT window <wdw> to page <page>
RW_REMOVE_WINDOW <wdw> *(on new line)* **<page>**	Remove D3PLOT window <wdw> from page <page>
AG_ADD_GRAPH <graph> *(on new line)* **<page>**	Add T/HIS graph <graph> to page <page>
RG_REMOVE_GRAPH <graph> *(on new line)* **<page>**	Remove T/HIS graph <graph> from page <page>

TEMPLATE

Read and write template files

The syntax is:

WRITE <filename>	Write a template file to <filename>
READ <filename>	Read a template file from <filename>

WRITE

Write data to the terminal

The syntax is:

<entity> <list> *(on new line)* **<component>**	Write <component> data for a <list> entities of type <entity>
SCAN <entity> <nn> *(on new line)* **<component>**	Display the maximum and minimum <nn> values of <component> for element type <entity>
GS_GLOBAL_SUMMARY <summary>	Display a model summary. Valid <summary>s are:

GS_GLOBAL_SUMMARY	Whole model summary
PS_PART_SUMMARY	Whole model summary by part
NRB_NODAL_RB_SUMMARY	Nodal rigid body summary
IS_INTERFACE_SUMMARY	Contact surfaces summary

COINCIDENT	Display coincident elements
OS_OUTPUT_TO_SCREEN	Switches screen output on/off
OF_OUTPUT_TO_FILE	Switches file output on/off

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

FILE_FORMAT <option>	Sets the output file format. Valid <option>s are:

TEXT	Text file format
CSV	CSV file format
XLSX	Excel XLSX file format

ENVELOPE <option>	Turn envelope on. Valid <option>s are:

<table>
<tr><td rowspan="8">COMPONENT <type></td><td colspan="2">Select the type of value to write. Valid <type>s are:</td></tr>
<tr><td>OFF</td><td>Turn envelope plotting off</td></tr>
<tr><td>MAX_VALUE</td><td>Plot max values</td></tr>
<tr><td>TIME_OF_MAX_VALUE</td><td>Plot time of the max values</td></tr>
<tr><td>MIN_VALUE</td><td>Plot min values</td></tr>
<tr><td>TIME_OF_MIN_VALUE</td><td>Plot time of the min values</td></tr>
<tr><td>ABS_VALUE</td><td>Plot absolute values</td></tr>
<tr><td>TIME_OF_ABS_VALUE</td><td>Plot time of the absolute values</td></tr>
<tr><td>STATES <list></td><td colspan="2">Give a <list> of state numbers to be used</td></tr>
</table>

KEYWORD <option>	Write initial keyword data <option>s are:

<entity> <list>	Select a <list> of entities of type <entity>

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

FILENAME <fname>	Output filename
(NO_)NODE_COORDS	Turn on/off output of nodal coordinates
(NO_)NODE_CONSTRAINTS	Turn on/off output of nodal constraints
(NO_)ELEM_TOPOLOGY	Turn on/off output of element topology
(NO_)INITIAL_STRESS	Turn on/off output of initial stress
(NO_)INITIAL_STRAIN	Turn on/off output of initial strain
(NO_)INITIAL_NODAL_VEL	Turn on/off output of initial nodal velocity
APPLY	Write data to the file
STATUS	Show the current settings
EXPLAIN	Further help

More info on valid <list> arguments can be found here

XY_DATA

Plot XY data

The syntax is:

NODES <list> *(on new line)* **<component>**	Plot <component> data versus time for nodes in <list>
SOLIDS <list> *(on new line)* **<component>**	Plot <component> data versus time for solids in <list>
BEAMS <list> *(on new line)* **<component>**	Plot <component> data versus time for beams in <list>
SHELLS <list> *(on new line)* **<component>**	Plot <component> data versus time for shells in <list>

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

THICK_SHELLS <list> *(on new line)* **<component>**	Plot <component> data versus time for thick shells in <list>
STONEWALLS <list> *(on new line)* **<component>**	Plot <component> data versus time for stonewalls in <list>
INTERFACES <list> *(on new line)* **<component>**	Plot <component> data versus time for interfaces in <list>
PARTS <list> *(on new line)* **<component>**	Plot <component> data versus time for parts in <list>
AIRBAGS <list> *(on new line)* **<component>**	Plot <component> data versus time for airbags in <list>
SURFACES <list> *(on new line)* **<component>**	Plot <component> data versus time for surfaces
GLOBAL <component>	Plot global <component> data versus time
COMP_XY <entity> <list> *(on new line)* **<component1>** *(on new line)* **<component2>**	Plot <component1> data versus <component2> over time for the <list> of <entity>s
COMP_LINE <entity> <list> *(on new line)* **<component1>** *(on new line)* **<component2>**	Plot <component1> data versus <component2> at a given time for the <list> of <entity>s
SORT <action>	Sort the COMP_LINE data points. Valid <action>s are:

NO_SORT	No sorting
LABEL	Sort by label ids
X_VALUE	Sort by data x value
Y_VALUE	Sort by data y value
BX_COORD	Sort by item basic x coordinate
BY_COORD	Sort by item basic y coordinate
BZ_COORD	Sort by item basic z coordinate
CX_COORD	Sort by item current x coordinate
CY_COORD	Sort by item current y coordinate
CZ_COORD	Sort by item current z coordinate

SET_INTERVALS <start> <interval> <finish>	Set the start time for the first state to be extracted, the time interval between states and the time of the last state to be extracted
SELECT_STATES <list>	Give a <list> of state numbers to be used
SHOW_TIMES	Show state times in .ptf file

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

PLOT	Turn XY plotting on or off
WRITE_CURVES	Turn curve file writing on or off
WRITE_CSV	Turn csv file writing on or off
CURVE_NAMES <type>	Change the default curve file names. Valid <type>s are:

GLOBAL	Global data curves
PART	Part data curves
AIRBAG	Airbag data curves
CONTACT	Contact data curves
ELEMENT	Element data curves
NODE	Node data curves
COMPOSITE	Composite curves
SECTION	Cut section curves

NUMBER_OF_CURVES <curves>	Set the maximum number of curves in a .cur file to <curves>
STATUS	Show the current settings
EXPLAIN	Further help

More info on valid <list> arguments can be found here

IMAGES

Output images and animations

Select what to capture with the syntax:

ALL_PAGES	Select all the pages to capture
CURRENT_PAGE	Select the current page to capture
ONLY_WINDOW <win_num>	Select window <wiun_num> to capture
WHITE_BACKGROUND <switch>	Capture images with white background ON or OFF

Select the image type with the syntax:

`CT_CONTINUOUS_TONE` Continuous tone (solid) contour plot

JPEG <filename>	Write the current image as a jpeg to <filename>
BMP_U8 <filename>	Write the current image as an uncompressed 8-bit bitmap to <filename>
BMP_C8 <filename>	Write the current image as a compressed 8-bit bitmap to <filename>
BMP_U24 <filename>	Write the current image as an uncompressed 24-bit bitmap to <filename>
PPM <filename>	Write the current image as a portable pixmap to <filename>
PNG_8BIT <filename>	Write the current image as an 8-bit png to <filename>
PNG_24BIT <filename>	Write the current image as a 24-bit png to <filename>
GIF <filename>	Write the current image as a gif to <filename>
MP4 <filename>	Write the current animation as an mp4 to <filename>
AGIF <filename>	Write the current animation as a gif to <filename>
AVI_MJPG <filename>	Write the current animation as a motion-jpeg to <filename>
AVI_U8 <filename>	Write the current animation as an uncompressed 8-bit motion-jpeg to <filename>
AVI_C8 <filename>	Write the current animation as a compressed 8-bit motion-jpeg to <filename>
AVI_U24 <filename>	Write the current animation as an uncompressed 24-bit motion-jpeg to <filename>
EXPORT_3D <type> <filename>	Sets the write format <type> and writes a compressed file for 3D viewer to <filename>
EXPORT_3D_UNCOMPRESSED <type> <filename>	Sets the write format <type> and writes an uncompressed file for 3D viewer to <filename>
DITHER <level>	Set the dithering level
FRAME_RATE <n>	Set to <n> number of frames per second for animations

CT_CONTINUOUS_TONE	Continuous tone (solid) contour plot	
	REPEAT <n>	Set to <n> number of repeats for animations
	QUALITY <%age>	Image quality of J/MPEG files
	STATUS	Show the current settings
LABEL		
	Display labels	
	The syntax is:	
	<entity> <list>	A <list> of <entity>s to label
	LABEL	Toggle on or off if entities are labelled with their node/element number
	PN_PART_NUMBER	Toggle on or off if entities are labelled with their part/surface number
	NE_NODES_ON_ELEMENT	Toggle on or off if nodes attached to an element are labelled
	EN_ELEMENTS_ON_NODE	Toggle on or off if elements attached to a node are labelled
	NC_NODAL_COORDINATES	Toggle on or off if nodes are labelled with their coordinates
	DATA_VALUE	Toggle on or off if nodes/elements are labelled with their current data value
	STATUS	Show the current settings
	EXPLAIN	Further help
	More info on valid <list> arguments can be found here	
UTILITIES		
	General utility commands	
	The syntax is:	
	COLOUR_OF_ENTITIES <entity> <list> ***(on new line)*** **<colour>**	Set the colour of a <list> of <entity> type to <colour>
	TRANSPARENCY <entity> <list> ***(on new line)*** **<%age>**	Set the percentage transparency of a <list> of <entity> type to <%age>
	MEASURE_DISTANCE <action>	Measure distances on screen. Valid <action>s are:

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

PP_POINT_TO_POINT	Distance between two screen points (cursor picked)
NN_NODE_TO_NODE	Distance between two nodes (cursor picked)
NO_NODE_TO_ORIGIN	Current node location (cursor picked)
NA_NODE_ANGLE	Angle between vectors n1n2 and n1n3 (cursor picked)
PA_POINT_ANGLE	Angle between vectors p1p2 and p1p3 (cursor picked)

TITLE_MODIFY <title> Change the model title to <title>

UNATTACHED_NODES List any unattached nodes

TARGET_MARKERS <action> Add nodal target markers. Valid <action>s are:

RADIUS <rad>	Define the radius of the marker
COLOUR <col1> <col2>	Define the two colours <col1> <col2> of the marker
CREATE <node id>	Create a marker on node <node id>
DELETE <node id>	Delete the marker from node <node id>
OFF	Turn target markers off
ON	Turn target markers on

FAILURE_LOGIC <action> How to handle failed elements. Valid <action>s are:

DS_DELETED_SWITCH	Toggle displaying deleted elements
DL_DELETED_LIST	List deleted elements
FH_HATCHING_SWITCH	Toggle hatching deleted elements
FC_FAILED_CONTOUR <colour>	Hatching colour
FL_FAILED_LIST	List failed elements

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

CRITERION	Define failure criterion

DATA_COMPONENTS List data components

CUSTOMISE_GRAPHICS <action> Graphic options. Valid <action>s are:

LW_LINE_WIDTH <pixels>	Define line width in pixels
M3D_3D_GRAPHICS	Switch to 3D mode
M2D_2D_GRAPHICS	Switch to 2D mode
UPDATE_LEVEL <level>	Define the update level
SOFT_CLIP_SWITCH	Turn the software clipping on or off
SP_SHOW_PROJECTION	Turn the box showing a representation of the projection and clipping planes on or off
WINDOW_SIZE <x> <y>	Set the window size

FILE_SKIP <n> Skips <n> missing .ptf files

FAMILY_SIZE <size> Set the file member size to <size> in MBytes

SETTINGS_FILE <action> Write or read a settings file. Valid <action>s are:

Write to <filename>
Read from <filename>

PROPERTIES_FILE <action> Write or read a properties file. Valid <action>s are:

WRITE <model> ***(on new line)* <filename>**	Write properties of <model> to <filename>

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

READ <model> ***(on new line)*** **<filename>**	Read properties from <filename> into <mode>

TOPAZ_FILE <To> <Tf> <Ti> <size> <header> <family> <filename>

Write a topaz file

Where:

<To> is the time offset

<Tf> is the time factor

<Ti> is the time interval

<size> is the family member size in MBytes

<header> sets whether to write a header of initial data to file and can be YES or NO

<family> is the family member to start at

<filename> is the name of the file to write to

STL_FILE <filename> <type>

Write an .stl file to <filename> as a file in the format <type>

Where:

<type> is either ASCII or BINARY

EXTERNAL <action>

Plot externally defined data. Valid <action>s are:

READ_FILE <filename>	Read from <filename>
ON	Switch data off
OFF	Switch data on

GLOBAL_FACTORS <filename> Read <filename> for global factors

MENU_ATTRIBUTES Display the menu attributes panel

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

PTF_COMPRESS <action> Generate a new set of ptf files with a subset of the data. Valid <action>s are:

OUTPUT_TYPE <type>	Set the output type to ORIGINAL or REORDERED
MODEL <model id>	Set the current model
FILENAME <filename>	Set the filename
PART <list>	Set a <list> of parts to output
STATES <list>	Set a <list> of states to output
FAMILY_SIZE <size>	Set the maximum fanily size (in KB)
APPLY	Write the ptf files
ALL_ON	Turn on all output
ALL_OFF	Turn off all output
(NO_)VELOCITIES	Turn on/off output of nodal velocities
(NO_)ACCELERATIONS	Turn on/off output of nodal accelerations
(NO_)TEMPERATURES	Turn on/off output of nodal temperatures
(NO_)SHELL_STRESS	Turn on/off output of shell and thick shell stress tensor
(NO_)PLASTIC	Turn on/off output of plastic strain
(NO_)SHELL_FORCES	Turn on/off output of shell forces and moments
(NO_)SHELL_THICKNESS	Turn on/off output of shell thicknesses
(NO_)SHELL_EXTRA	Turn on/off output of shell extra variables
(NO_)SOLID_EXTRA	Turn on/off output of solid extra variables
(NO_)BEAM_EXTRA	Turn on/off output of beam extra variables
(NO_)STRAIN	Turn on/off output of strain tensor
(NO_)SPH_STRESS	Turn on/off output of strain tensor for SPH elements

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

(NO_)SPH_PLASTIC	Turn on/off output of plastic strain for SPH elements
(NO_)SPH_STRAIN	Turn on/off output of stress tensor for SPH elements
(NO_)AIRBAG	Turn on/off output of airbag particle data
(NO_)SOLID_STRESS	Turn on/off output of stress tensor for solid elements (reordered database only)
(NO_)SOLID_PLASTIC	Turn on/off output of plastic strain for solid elements (reordered database only)
(NO_)SOLID_VM_STRESS	Turn on/off output of Von Mises stress for solid elements (reordered database only)
(NO_)SHELL_VM_STRESS	Turn on/off output of Von Mises stress for shell elements (reordered database only)
(NO_)TSHELL_VM_STRESS	Turn on/off output of Von Mises stress for thick shell elements (reordered database only)
(NO_)SOLID_VM_STRAIN	Turn on/off output of Von Mises strain for solid elements (reordered database only)
(NO_)SHELL_VM_STRAIN	Turn on/off output of Von Mises strain for shell elements (reordered database only)
(NO_)TSHELL_VM_STRAIN	Turn on/off output of Von Mises strain for thick shell elements (reordered database only)
(NO_)SHELL_ENG_STRAIN	Turn on/off output of Engineering strain for shell elements (reordered database only)
(NO_)SPOTWELD	Turn on/off output of Spotweld data (reordered database only)
(NO_)SPC	Turn on/off output of SPC data (reordered database only)
(NO_)SPRING	Turn on/off output of Spring data (reordered database only)
(NO_)SEATBELT	Turn on/off output of Seatbelt data (reordered database only)
(NO)_NODES_FOR_ZTF	Turn on/off output of nodes for ZTF items

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

(DONT)_EMBED_ZTF	Do/Don't embed ztf file in database (reordered database only)

LC_COMBINATION <action> Combine Nastran linear static subcases. Valid <action>s are:

MODEL <model id>	Set the current model
SUBCASE <subcase id> <factor>	Add <subcase id> to the list of subcases to be combined
NAME <name>	Set the combined loadcase name
APPLY	Create the combined loadcase with the currently selected subcases
READ <filename>	Read a loadcase combination file
WRITE <filename>	Write a loadcase combination file
RESET	Clear the selected subcases

USER_DEFINED_NAME <action> Define names for entities. Valid <action>s are:

MODEL <model id>	Set the current model
ADD <entity> <entity id> ***(on new line)*** **<name>**	Define a <name> for entity type <entity>, id <entity id>
DELETE <entity> <entity id>	Delete a defined name for entity type <entity>, id <entity id>
NAME_SWITCH <entity> <action>	Toggle the display of names for enity type <entity>
READ <filename>	Read a user defined names file
WRITE <filename>	Write a user defined names file

More info on valid <list> arguments can be found here

GROUPS

Group options

The syntax is:

CREATE Create a group

ADD <entity> <list>	Add the <list> of entities of type <entity> to the current group

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

REMOVE <entity> <list>	Remove the <list> of entities of type <entity> from the current group
CLEAR	Clear the current group
STORE	Store the current group **NEW <num> <name>** — Store it as group number <num> and <name> **EXISTING <group>** — Store it in existing <group>
LIST	List the contents of the current group

SKETCH <group>	Sketch the group. <group> can be the number or name of the group.
MODIFY <group>	Modify a stored group. <group> can be the number or name of the group.

ADD <entity> <list>	Add the <list> of entities of type <entity> to the current group
REMOVE <entity> <list>	Remove the <list> of entities of type <entity> from the current group
CLEAR	Clear the current group
STORE	Store the current group **NEW <num> <name>** — Store it as group number <num> and <name> **EXISTING <group>** — Store it in existing <group>
LIST	List the contents of the current group

RENAME <group>	Rename a stored group. <group> can be the number or name of the group.

CT_CONTINUOUS_TONE Continuous tone (solid) contour plot

NAME <name>	New name
NUMBER <num>	New number
APPLY	Apply the new name and/or number

DELETE <group>	Delete a stored group. <group> can be the number or name of the group.
SAVE <filename>	Save the stored groups as an ascii file to <filename>
READ <filename>	Read the ascii file <filename>
READ_OLD <filename>	Read an old style binary group file <filename>
LIST	List the stored groups

More info on valid <list> arguments can be found here

JAVASCRIPT

JavaScript interface

The syntax is:

COMPILE <filename>	Compiles the JavaScript <filename> to check it for errors
EXECUTE <filename>	Executes the most recently compiled javaScript file or compiles and executes <filename> if given and different
MEMORY <size>	Sets JavaScript memory arena to <size> in MBytes

STATUS	Display programme status
NEW_MODEL <filename>	Open model <filename>
THF	For internal use with D3PLOT-T/HIS linked sessions. Manually entering commands within this menu is not recommended and is typically only used for automatic private use by the respective programs.
EXIT	Exit D3PLOT

20.5.3. Global Menu Commands

Global Menu Commands:

`DRAW`	Draws the undeformed geometry in line mode
`LINE`	Draws the current state in line mode
`HIDDEN_LINE`	Draws the current state in hidden-line mode
`SHADED`	Draws the current state in shaded mode
`REDRAW`	Redraws all windows
`CW_CURRENT_WINDOW <window id>`	Sets the current window to <window id>
`CM_CURRENT_MODEL <model id>`	Sets the current model to <model id>
`SXY, SYZ, SZX`	Preset views on XY, YZ and XZ axes
`+XY, +YZ, +ZX, +XZ`	Alternative syntax for the above
`-XY, -YZ, -ZX, -XZ`	Reversed views of the above
`ISOMETRIC`	Preset isometric view
`+ISO, -ISO`	Same as above and reverse
`RS <x y z>`	Rotate about screen coordinates <x y z>
`RM <x y z>`	Rotate about model coordinates <x y z>
`TR <x y>`	Translate along screen axis <x y>
`ZM`	Zoom in using cursor
`CENTRE`	Centre image on cursor position
`MG <scale>`	Magnify by factor <scale>
`AU_AUTOSCALE_UNDEF`	Autoscale on undeformed geometry
`AC_AUTOSCALE_CURR`	Autoscale on current geometry
`ZERO_VIEW`	Reset view to plan on XY plane
`TIME <time>`	Set to a state at time <time> Where: <time> is a time value, PREVIOUS, NEXT or LAST

`STATE <number>`	Set to a state by <number> Where: <number> is a state number, PREVIOUS, NEXT or LAST
`FILE_SCAN`	Re-scan file family for new states
`SS_SHOW_STATES`	Show all state times in the file
`GM_GLOBAL_MENU`	Display global menu command summary
`GH_GLOBAL_HELP`	Help on global commands
`GE_GLOBAL_EXPLAIN`	Details of global commands

20.6. F. NASTRAN OP2 File

F. NASTRAN OP2 FILE

From D3PLOT 11.0 onwards, you can read in results from NASTRAN OP2 files assuming the input file has been run with the case control commmand;

PARAM,POST,-2

D3PLOT requires that the OP2 file contains node and element information so the input file should **NOT** have the case control command:

PARAM,OGEOM,NO

The following is a list of what is currently read from the OP2 file.

20.6.1. Solution Types

Solution Types

The following solution types can be read in to D3PLOT.

- SOL 101 - Linear Statics
- SOL 103 - Normal Modes
- SOL 106 - Non-Linear Statics
- SOL 107 - Direct Complex Eigenvalues
- SOL 108 - Direct Frequency Response
- SOL 109 - Direct Transient Response
- SOL 110 - Modal Complex Eigenvalues
- SOL 111 - Modal Frequency Response
- SOL 112 - Modal Transient Response

Other solution types may read in to D3PLOT, but have not been tested.

20.6.2. Elements

Elements

The following elements can be read in to D3PLOT.

NASTRAN ELEMENTs plotted as **SOLID** s

- **CHEX20**
- **CHEX8**
- **CHEXA**
- **CHEXAFD**

- **CHEXA20F**
- **CHEXPR**
- **CHEXAL**
- **CHEXA1**
- **CHEXA2**
- **CPENTA**
- **CPENPR**
- **CPENT15F**
- **CPENT6FD**
- **CTETRA**
- **CTETPR**
- **CTETR4FD**
- **CTETR10F**
- **CWEDGE**

NASTRAN ELEMENTs plotted as **BEAM** s

- **CBAR**
- **CBEAM**
- **CONROD**
- **CROD**
- **CTUBE**
- **PLOTEL**

NASTRAN ELEMENTs plotted as **SHELL** s

- **CQUAD**
- **CQUAD1**
- **CQUAD2**
- **CQUAD4**
- **CQUAD4FD**
- **CQUADR**
- **CQUAD8**
- **CQUADX**
- **CQUAD9FD**
- **CTRBSC**
- **CTRIA1**
- **CTRIA2**
- **CTRIA3**
- **CTRIA6**
- **CTRIAR**
- **CTRIARG**
- **CTRIA3FD**
- **CTRIA6FD**
- **CTRIAX**
- **CTRIAX6**

NASTRAN ELEMENTs plotted as **SPRING** s

- **CBUSH**
- **CBUSH1D**
- **CBUSH2D**
- **CDAMP1**
- **CDAMP2**
- **CELAS1**
- **CELAS2**
- **CELAS3**
- **CELAS4**
- **CGAP**
- **CMASS1**
- **CMASS2**
- **CVISC**

20.6.3. Data Components

Data Components

The data components that will be available in D3PLOT will depend on which case control output statements were in the input file. The following statements are supported:

DISPLACMENT(PLOT)
VELOCITY(PLOT)
ACCELERATION(PLOT)
THERMAL(PLOT)

STRESS(PLOT)
STRAIN(PLOT)
FORCE(PLOT)
ESE(PLOT)
EKE(PLOT)
EDEL(PLOT)

SPCFORCE(PLOT)

Nodal Data

The following components will be available if the **DISPLACEMENT** command was used:

DX_X_DISPLACEMENT	**RDX_X_ROTATION**

DY_Y_DISPLACMENT	**RDY_Y_ROTATION**
DZ_Z_DISPLACEMENT	**RDZ_Z_DROTATION**
DR_DISP_RESULTANT	**RDR_ROT_RESULTANT**

The following components will be available if the **VELOCITY** command was used:

VX_X_VELOCITY	**RVX_X_ROTATION**
VY_Y_VELOCITY	**RVY_Y_ROTATION**
VZ_Z_VELOCITY	**RVZ_Z_ROTATION**
VR_VEL_RESULTANT	**RVR_ROT_RESULTANT**

The following components will be available if the **ACCELERATION** command was used:

AX_X_VELOCITY	**RAX_X_ROTATION**
AY_Y_VELOCITY	**RAY_Y_ROTATION**
AZ_Z_VELOCITY	**RAZ_Z_ROTATION**
AR_ACCEL_RESULTANT	**RAR_ROT_RESULTANT**

The following components will be available if the **THERMAL** command was used:

TEMPERATURE

Element Data

The following components will be available if the **STRESS** command was used:

SOLID and **SHELL** elements:

X_DIRECT_STRESS	**XY_SHEAR_STRESS**
Y_DIRECT_STRESS	**YZ_SHEAR_STRESS**
Z_DIRECT_STRESS	**ZX_SHEAR_STRESS**

The following components will be available if the **STRAIN** command was used:

SOLID and **SHELL** elements:

SX_DIRECT_STRAIN	**SXY_SHEAR_STRAIN**
SY_DIRECT_STRAIN	**SYZ_SHEAR_STRAIN**
SZ_DIRECT_STRAIN	**SZX_SHEAR_STRAIN**

BEAM elements:

SAX_AXIAL_STRAIN (**Note** : For CBEAM elements this is the Longitudinal strain at point C).

The following components will be available if the **FORCE** command was used:

SHELL elements:

FX_NORMAL_FORCE	**MX_BENDING_MOMENT**	**QXZ_SHEAR_FORCE**
FY_NORMAL_FORCE	**MY_BENDING_MOMENT**	**QYZ_SHEAR_FORCE**
FXY_SHEAR_FORCE	**MXY_BENDING_MOMENT**	

BEAM elements:

FX_AXIAL_FORCE	**MXX_TORSIONAL_MOMENT**
FY_Y_SHEAR_FORCE	**MYY_BENDING_MOMENT**
FZ_Z_SHEAR_FORCE	**MZZ_BENDING_MOMENT**

SPRING elements (**Note** : results are plotted on beams):

FX_AXIAL_FORCE

The following components will be available if the **ESE** command was used:

SOLID and **SHELL** elements:

SEN_STRAIN_ENERGY

SENP_STRAIN_ENERGY_PERCENT

SEND_STRAIN_ENERGY_DENSITY

BEAM elements:

BSEN_STRAIN_ENERGY

BSENP_STRAIN_ENERGY_PERCENT

BSEND_STRAIN_ENERGY_DENSITY

The following components will be available if the **EKE** command was used:

SOLID and **SHELL** elements:

KEN_KINETIC_ENERGY

KENP_KINETIC_ENERGY_PERCENT

KEND_KINETIC_ENERGY_DENSITY

BEAM elements:

BKEN_KINETIC_ENERGY

BKENP_KINETIC_ENERGY_PERCENT

BKEND_KINETIC_ENERGY_DENSITY

The following components will be available if the **EDEL** command was used:

SOLID and **SHELL** elements:

ENL_ENERGY_LOSS

ENLP_ENERGY_LOSS_PERCENT

ENLD_ENERGY_LOSS_DENSITY

BEAM elements:

BENL_ENERGY_LOSS

BENLP_ENERGY_LOSS_PERCENT

BENLD_ENERGY_LOSS_DENSITY

SPC Data

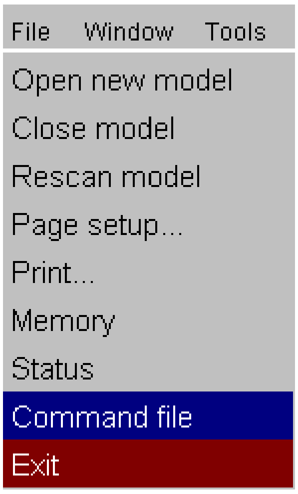
The following components will be available if the **SPCFORCE** command was used:

SPC_R_FORCE(X)	**SPC_R_MOMENT(X)**
SPC_R_FORCE(Y)	**SPC_R_MOMENT(Y)**
SPC_R_FORCE(Z)	**SPC_R_MOMENT(Z)**
SPC_R_FORCE(MAG)	**SPC_R_MOMENT(MAG)**

Note that you will need to have a d3plot.components file in the installation directory for these components to be available.

20.7. G. Command Files

G. COMMAND FILES



20.7.1. Introduction to Command Files

Introduction to Command Files

It is possible to record commands issued during a D3PLOT session to a file and replay these during the current or a subsequent session.

This is exactly equivalent to having two tape-recorders available: one to record commands and one to play them back. It is also possible to have both turned on at the same time so that commands being played back from an earlier session get recorded (and possibly edited) during the current session.

Note that Settings Files may offer a more convenient alternative to command files, if the aim is to generate a particular plot.

20.7.2. How Screen Menu Events are Recorded

How screen-menu events are recorded

In command-line mode the files are simple: they simply record and play back verbatim typed in commands. When screen-menu mode is in use it becomes necessary to record commands expressed as button clicks, slider motions, etc; so a different strategy is required.

D3PLOT stores these screen menu "events" in special codes in the file, and replays them as if they had been applied by the user.

20.7.3. Compatibility with Files from Earlier Releases of D3PLOT

Compatibility with Files from Earlier Releases of D3PLOT

Since command-line syntax has been preserved in this version, and command input into the dialogue window is still permitted, command-line command files are backward-compatible. Button-click command files are NOT backward-compatible.

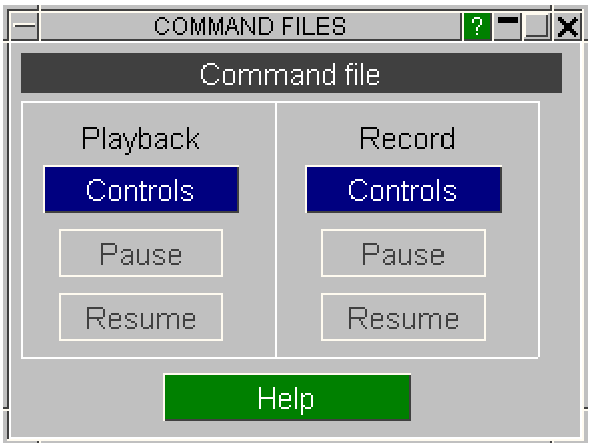
The file format used for Oasys Suite 7.0 command files is itself backwards compatible: you should not need to modify existing files at all for them to work with this version.

20.7.4. CFILE Invoking the Command-File Launcher Box

CFILE Invoking the Command-File Launcher Box

Recording and replaying files are controlled from separate windows that are "launched" from the "Command Files" launcher box.

This can also be used during the operation of these files, see here .



20.7.5. Recording Files

Recording files

This figure (right) shows the **RECORD** control panel.

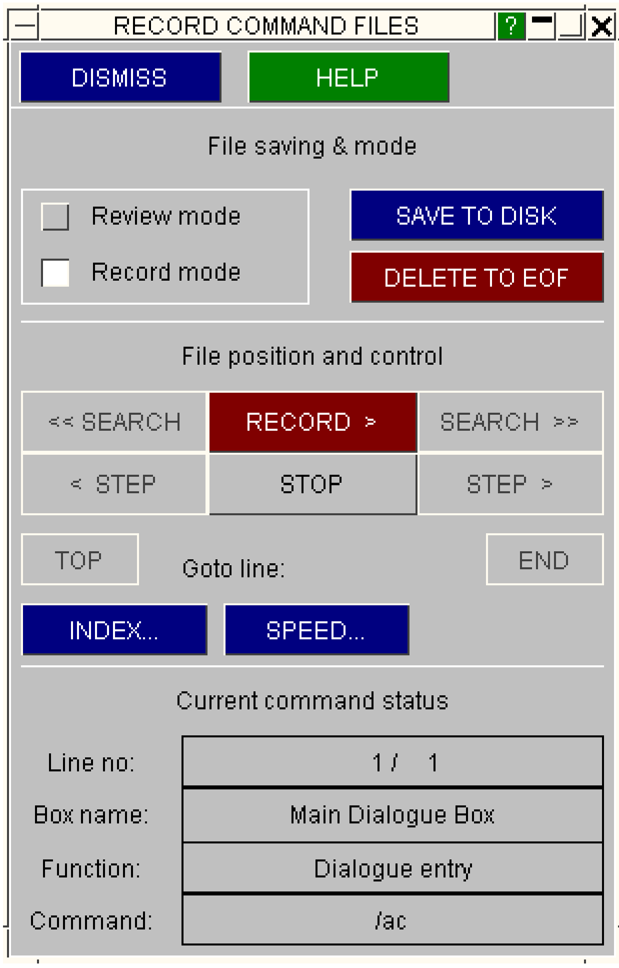
The **[RECORD] CONTROLS...** button in the launcher box maps this panel, which manages the recording of commands.

It works exactly like a tape-recorder: you can use the **RECORD** button to switch on recording, and all subsequent commands are recorded until it is turned off again by **STOP** .

It has two modes:

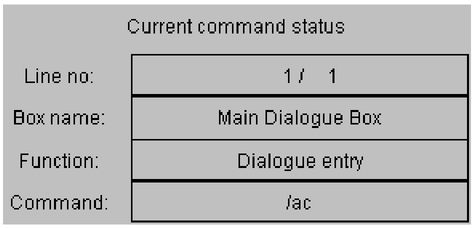
Record Actually records commands. This is the default.

Review Allows you to go back and forth to review recorded commands.



How files are recorded

When in **Record** mode, with the **RECORD** button depressed, every command, screen-pick, button press, etc (except those in the command file control panels) is recorded in an internal file. You could think of this as being the tape of a video recorder, with each video "frame" representing a command.



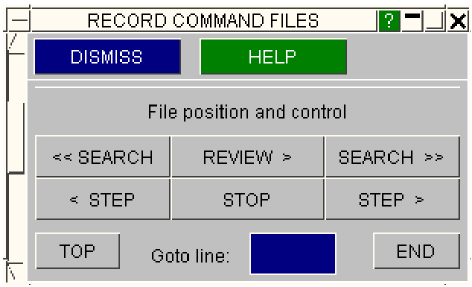
As each command is stored it is also reported in the **Current command status** area at the base of the box.

Each command ha a unique line number, and in this example line #1 (out of 1 line recorded so far) is the command `/ac` in the dialogue box.

Reviewing stored commands

If you switch from **Record** mode to **Review** mode then you will find that all the tape recorder buttons become live, and that (implicitly) command recording stops. You will also note that the **RECORD** button becomes a **REVIEW** button:

In review mode you can list stored commands using the commands shown here.



REVIEW > Plays the recorded commands forwards from the current position, listing them in the **Current command status** box as described above. A time delay of 1 second between commands is left by default, but you can alter this using the **SPEED...** options. **STOP** halts this process.

STEP < & > Respectively step backwards and backwards a single command.

SEARCH << & >> Respectively search backwards and forwards for a specified command.

You can search for any permutation of box name, function or dialogue string.

The options shown here are presented to you when you use either **SEARCH** button. It is recommended that you use the [**?**] buttons to identify boxes or functions from menus, as these will get the syntax right. Command string searches are not case sensitive.

TOP & END Will take you to line #1 and the last line respectively.

Goto line: Will take you to the line number you specify.

Editing and overwriting commands

When you switch back into **Record** mode and start recording commands again, they start being recorded at the current position in the file. So if you have moved backwards during a **Review** , then start recording, you will overwrite commands stored at that point - just as would happen in an ordinary tape-recorder.

Commands are overwritten on a line-by-line basis. Consider this example:

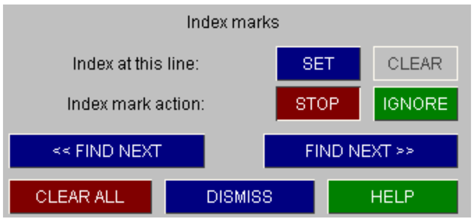
- You record 20 commands.
- You go back to line #13 in review mode.
- You record 4 commands starting at line #13.

In this situation the original four commands (#13 to #16) will be overwritten, but commands #17 to #20 will be unchanged.

To delete commands from the current position to end of file use the **DELETE TO EOF** button.

It is possible to edit command files using a text editor. WE recommend using command-line commands where these are available. These can be inserted in the file amongst button-click commands. To find out about the command-line menu system, type H or M in the dialogue box.

Inserting index marks



You can put markers, "index marks", on any line in your file. These have no command significance, but are useful as targets for search operations. The **INDEX...** command options are shown here. Index marks can be **SET** and **CLEAR** ed, and you can search for them in either direction with the **FIND NEXT << & >>** buttons.

Writing out files

Files are stored as internal scratch files, and must be written out to disk in ASCII form before they can be read back in. This is done by the **SAVE TO DISK** command.

WARNING: **If you record a file, and then exit D3PLOT without issuing the SAVE_TO_DISK command, your recorded commands will be lost.**

The format of the ASCII file, and advice on editing it, is given here .

20.7.6. Playing Back Files

Playing back files

This figure shows the command file playback control panel.

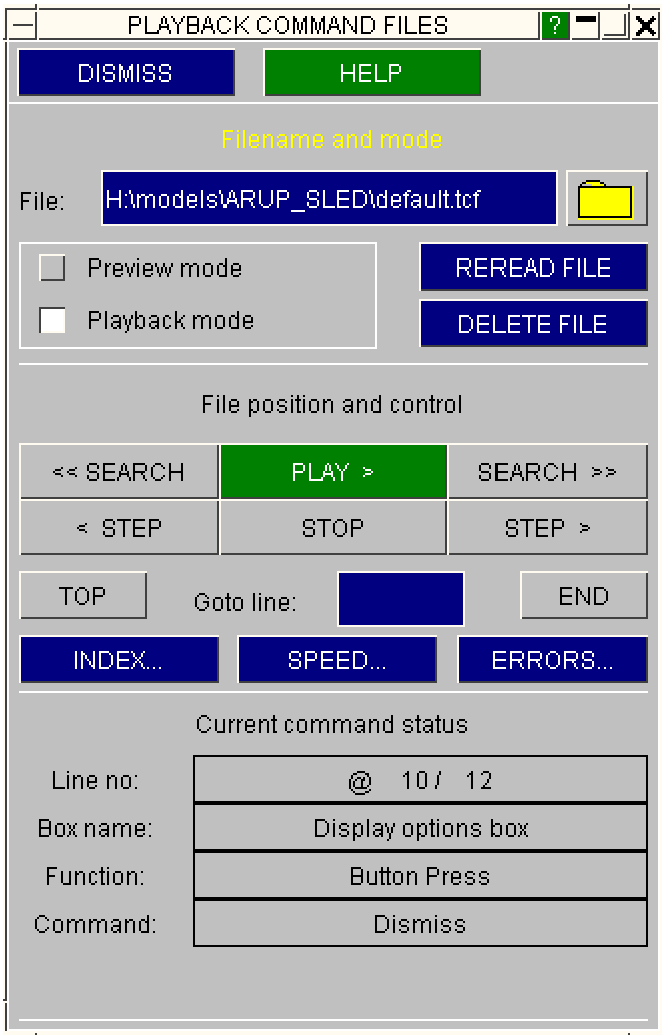
To use it you must first enter a command filename in the **File:** box. This is read in and converted into internal form, and the command buttons will then become live.

To execute commands continuously you use the **PLAY >** button. This can be halted with the **STOP** button. **STEP** executes commands one at a time.

It has two modes:

Playback Executes commands. This is the default.

Preview Previews commands without executing them.



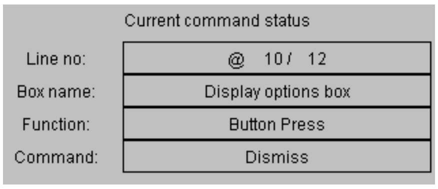
How command files are processed

Commands are stored in an internal scratch file that, as with "record" mode, is like a tape in a tape recorder. Each command forms a separate line, and you can move the "tape" back and forth to locate it at any line.

When in **Playback** mode you can **PLAY** your commands. Each command from the current position to <end of file> is executed as if you had typed it in, screen-picked it, etc. To stop a playback prematurely use the **STOP** button.

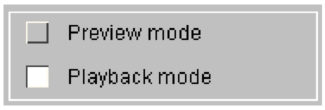
By default a **PLAY** operation operates at full speed, but you can introduce a delay between commands using the **SPEED...** options. This can be useful if you want to interrupt a sequence at a particular point, and if full speed playback is too fast to follow.

The current command is always shown in the **Current command status** box. In the example here we are currently at command #10 (out of 12 in the file), which is a **DISMISS** button press.



Previewing commands

The playback control panel can be operated in **Preview** (instead of **Playback**) mode. This allows you to view commands without actually executing them, and thus to position yourself where you want in the file.



The **PLAY >** button is replaced by a **PREVIEW >** one to remind you which mode you are in. Commands are listed in the **Current command status** box as above, and you can **STEP** backwards and forwards as before. These and the other positioning and searching commands operate in exactly the same way as for recording files: see here for more detailed instructions.

Handling errors during playback

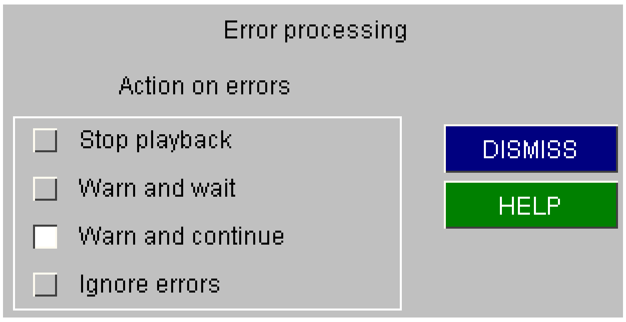
When you are playing commands it is possible to generate errors. Examples might be playing back commands recorded in a model with 10 materials in a different one with only five materials. If the screen button corresponding to the (non-existent) 10th material is picked in some context (eg from a menu) an error will occur.

The **ERRORS...** command lets you set the action to be taken when errors occur.

The **Action on errors** options are shown here, with the default **Warn and continue** option set.

The other options are self-explanatory.

This gives more information about how errors may occur, and how to avoid them.



Opening new command files

You can read in a new command file at any time, or reread the current one (assuming it has been updated on disk).

In either case **all** existing commands in the internal scratch file are deleted and superseded by those read in. You cannot concatenate the contents of the new file with the existing one.

Recording files during command file playback

This is quite legal. The two operations are quite separate, and commands executed from a command file will be recorded exactly as if they had been typed in. This is the recommended method of editing and appending to files.

20.7.7. Using the "Launcher" Box During Recording and Playback

Using the "Launcher" Box During Recording and Playback

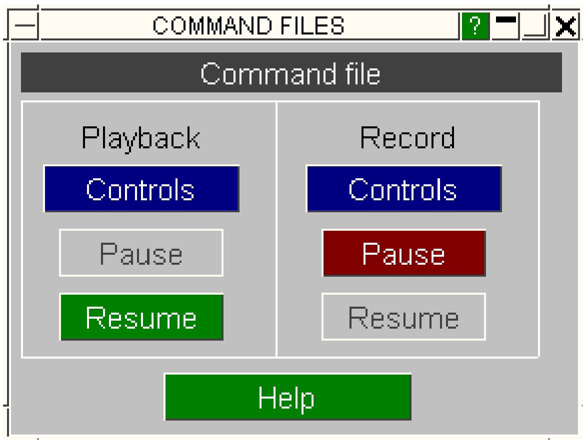
This figure shows the "launcher" box part of the way through a simultaneous record and playback session.

This box need not be mapped, but it can be useful to use it rather than the full control panels since it uses less space.

The "launcher" box always echos the **PLAYBACK/STOP** status of the playback box if a command file is resident in memory.

It also echos the **RECORD/STOP** status of the record box if that is in use.

Since the main panels for these two operations take up a lot of space on the screen you can use the simple on/off functionality provided by the "launcher" box instead.



Think of it as a primitive remote control that can only has a "pause" function. It operates as follows:

During Playback: When you are in **PLAY** mode the **PAUSE** button will be available, and you can press it to pause playback.

Once pressed, as in the example above, it will be greyed out and the **RESUME** button made available instead. This, obviously, continues the playback.

During Recording: When **RECORD** is on, as in the example here, the **PAUSE** button is live. If pushed it suspends recording, and subsequent commands will not be recorded until **RESUME** is pressed.

In either case you can of course invoke the main control panels and override these buttons.

When recording or playback are not active then the buttons on this panel will be greyed out: they only become "live" when they might be needed.

20.7.8. More Information About Command Files

More Information About Command Files

The format of the ASCII disk files used for these

The ASCII files are up to 132 columns wide, with column numbers reserved as follows:

1 to 80

Dialogue strings from the dialogue box, button text, etc

81

Continuation character for long strings (`c` below)

82 to 132

Encoded screen-menu functions

Diagrammatically this gives:

<Dialogue string (80 characters)> |C| <Encoded screen-menu function values>

If the continuation column (#81) is occupied then it assumed to be a "long" line, and the character string can extend to column 132. Any screen-menu function values will then appear on the next line.

If columns 82 to 132 are empty the command is assumed to be a simple "dialogue" command, typed in by the user.

An example of a command file is:

```
$
$ D3PLOT version 7.0
session file
$ Date/time: 25-Nov-
95 17:42:00
$
/re 2                 1     1     1     0     0     0     0     0
/bl sta               1     1     1     0     0     0     0     0
/hi                   1     1     1     0     0     0     0     0
DISMISS               26    3     2     1     0     0     0     0
CFILE                 23    3     2     14    0     0     0     0
```

Backwards compatibility with command files from earlier versions of D3PLOT

This file format is fully backwards compatible with text-only command files from earlier versions of D3PLOT. They will have nothing in columns 82 to 132, and so will be treated as command-line input, which is what they are.

The command-line syntax in Version 7.0 of D3PLOT is virtually unchanged from earlier releases, so the vast majority of existing command-files should run with no modification.

Editing command files by hand

You can type in commands (in columns 1 to 80) that are genuine "command-line" commands, and you should leave columns 82 to 132 blank.

It is also possible to concatenate screen-menu command files using the editor, **BUT** beware the following:

Context errors: The instructions "drive along motorway", and "remove wheel nuts and take wheel off car" are both perfectly sensible. But only if the intervening commands "in case of puncture pull over to hard shoulder, stop, and jack up car" are inserted.

If you concatenate two sets of command files you can generate context errors if you miss out intervening steps. For example if file one sets up contour levels, and file two operates animation controls, you will have problems because you have omitted to switch from "static" to "animation" modes in between.

This is a context error. Normally you would not be able to operate animation controls while in "static" mode, since the buttons would not be available. But this protection is absent when replaying command files.

Range errors: Menus of items, for example lists of materials, may have fewer rows in this file than in the one where the command file was generated.

Since the screen-menu function recorded is the menu row, not its contents, this produces errors if you try to reference rows for non-existent entries.

To concatenate files it is better to **PLAY** each of them in turn through D3PLOT, while simultaneously **RECORD** ing them. This will allow you to spot and fix context errors more quickly, and also to insert missing commands if needed.

Certain commands are not recorded

Commands in "pop-up" windows, for example warning, confirmation and listing boxes are not recorded.

Also commands in the file filter box are not. You can make this work for you: if you want to define a different filename at playback time use [**?**] for the file filter box, if you want to use a fixed filename without intervention at playback time type it explicitly into the **File:** text box as this will be recorded.

20.7.9. Associating Command Files with Function Keys

Associating command files with Function keys

From D3PLOT 8.3 onwards the keyboard function keys **F1** to **F12** can be programmed to play back command files. This is described under **UTILITIES, FUNCTION KEYS** in UTILITIES, FUNCTION KEYS .

20.7.10. Running Command Files from the Command Line

Running command files from the command line

When running D3PLOT in batch it can be convenient to invoke a command file to run a specified set of commands. The following command-line arguments may be used to do this:

`-cf=` *<command filename>*	Runs "command filename" until its end, and then reverts to normal command line input.
`-exit`	This optional. If used then D3PLOT exits when the end of the command file is reached.

More information about command-line syntax is given in Appendix C .

21. Installation Organisation

Installation organisation

The Oasys Suite 21.1 installation can be customised to try and avoid a number of issues that often occur in large organisations with many users.

- Large organisations generally imply large networks, and it is often the case that the performance of these networks can be intermittent or poor, therefore it is common practice to perform an installation of the software on the local disk of each machine, rather then having a single installation on a remote disk.

  This avoids the pauses and glitches that can occur when running executable files over a network, but it also means that all the configuration files in, or depending upon, the top level "Admin" directory have to be copied to all machines and, more to the point, any changes or additions to such files also have to be copied to all machines.

- In larger organisations the "one person per computer" philosophy may not apply, with the consequence that users will tend to have a floating home area on a network drive and may not use the same machine every day.

  This is not usually a problem on Linux where the "home" directory is tied to the login name not the machine. However on Windows platforms it means that %USERPROFILE%, which is typically on the local C drive of a machine, is not a good place to consider as "home" since it will be tied to a given computer, therefore a user who saves a file in their home directory on machine A may not be able to access it from machine B.

- In a similar vein placing large temporary files on the /tmp partition (Linux) or the C: drive (Windows) may result in local disks becoming too full, or quotas exceeded.

This section gives only a brief summary of the installation organisation, and you should refer to the separate Installation Guide if you want to find out more about the details of installation, licensing, and other related issues.

21.1. Oasys Suite 21.1 Installation Structure

Oasys Suite 21.1 Installation structure

In Oasys Suite 21.1 the option is provided to separate a top-level 'administration' directory from the 'installation' one where the executables are located.

For large installations on many machines this allows central configuration and administration files to exist in one place only, but executables to be installed locally on users' machines to give better performance. Oasys Suite 21.1 also allows the following items to be configured

- The location for user manuals and other documentation.
- The definition of a user's home directory.
- The definition of the temporary directory for scratch files.

In addition parsing of the 'oa_pref' (preferences) file will now handle environment variables, so that a generic preference can be configured to give a user-specific result, and preferences may be 'locked' so that those set at the administration level cannot be changed by users.

These changes are entirely optional, and users performing a simple installation on a single machine do not need to make any changes to their existing installation practice.

Directory	**Status**	**Directory Content and purpose**	**oa_pref file option**
OA_ADMIN_xx	*Optional*	Top level configuration files. (**xx** =21 for Oasys Suite 21.1, thus **OA_ADMIN_21**) Admin level oa_pref file Other configuration files Timeout configuration file	
OA_ADMIN	*Optional*	Same as **OA_ADMIN_21**, provided for backwards compatibility with earlier releases. It is recommended that plain **OA_ADMIN**, without the **_xx** version suffix, is not used since otherwise there is no easy way of distinguishing between parallel installations of different releases of the Oasys Ltd software in an installation. *If **OA_ADMIN_21** is not defined then this non-release specific version is checked.*	
OA_INSTALL_xx	*Optional*	(**xx** =21 for release 21.1, thus **OA_ADMIN_21**	**oasys*install_dir:** ***<pathname>***

		All executables Installation level oa_pref file	
OA_INSTALL	*Optional*	Same as **OA_INSTALL_21**. If no "**OA_ADMIN_xx**" directory is used and all software is simply placed in this "install" directory, which would be typical of a single-user installation, then it is recommended that the **_xx** version suffix is used in order to keep parallel installations of different releases of the Oasts Ltd software separate on the machine. *If **OA_INSTALL_21** is not defined then this non-release specific version is checked*	**oasys*install_dir:** **<*pathname*>**
OA_MANUALS	*Optional*	Specific directory for user manuals. If not defined then will search in: **OA_ADMIN_xx/manuals** (xx = major version number) **OA_INSTALL/manuals**	**oasys*manuals_dir:** **<*pathname*>**
OA_HOME	*Optional*	Specific "home" directory for user when using Oasys Ltd software. If not defined will use: **$HOME** (Linux) **%USERPROFILE%** (Windows)	**oasys*home_dir:** **<*pathname*>**
OA_TEMP	*Optional*	Specific "temporary" directory for user when using Oasys Ltd software. If not defined will use: **P_tmpdir** (Linux, typically /tmp) **%TEMP%** (Windows, typically C:\temp)	**oasys*temp_dir:** **<*pathname*>**

It will be clear from the table above that no Environment variables have to be set, and that all defaults will revert to pre-Oasys Suite 9.4 behaviour. In other words users wishing to keep the status quo will find behaviour and layout unchanged if they do nothing.

OA_INSTALL_XX

Previously the software used the **OA_INSTALL** (renamed from **OASYS**) environment variable to locate the directory the software was installed in.

- On Windows this is no longer required as the software can work out its own installation directory. As this environment variable is no longer required it is recommended that it is removed from machines it is currently set on as in some cases where more than one version has been installed in different directories it can cause problems.

- On LINUX systems the "oasys_21" script that starts the SHELL automatically sets this Environment Variable and passes it to any application started from the SHELL. If you run applications directly from the command line and bypass the SHELL then you should set **`OA_INSTALL_XX`** so that the software can locate manuals and other required files.

`OA_ADMIN_XX`
Users wishing to separate configuration and installation directories will be able to do so by making use of the new top level **`OA_ADMIN_xx`** directory.

21.1.1. Installation Examples

Installation organisation

The Oasys Suite 21.1 installation can be customised to try and avoid a number of issues that often occur in large organisations with many users.

- Large organisations generally imply large networks, and it is often the case that the performance of these networks can be intermittent or poor, therefore it is common practice to perform an installation of the software on the local disk of each machine, rather then having a single installation on a remote disk.

  This avoids the pauses and glitches that can occur when running executable files over a network, but it also means that all the configuration files in, or depending upon, the top level "Admin" directory have to be copied to all machines and, more to the point, any changes or additions to such files also have to be copied to all machines.

- In larger organisations the "one person per computer" philosophy may not apply, with the consequence that users will tend to have a floating home area on a network drive and may not use the same machine every day.

  This is not usually a problem on Linux where the "home" directory is tied to the login name not the machine. However on Windows platforms it means that %USERPROFILE%, which is typically on the local C drive of a machine, is not a good place to consider as "home" since it will be tied to a given computer, therefore a user who saves a file in their home directory on machine A may not be able to access it from machine B.

- In a similar vein placing large temporary files on the /tmp partition (Linux) or the C: drive (Windows) may result in local disks becoming too full, or quotas exceeded.

This section gives only a brief summary of the installation organisation, and you should refer to the separate Installation Guide if you want to find out more about the details of installation, licensing, and other related issues.

21.1.2. Dynamic Configuration Using the Top Level oa_pref File

Dynamic configuration using the top level oa_pref file.

A further improvement is that all environment variables below **`OA_ADMIN_xx`** may either be set explicitly, or dynamically using the options in the oa_pref file at the top **`OA_ADMIN_xx`** level. This permits parallel installations of different versions of the software to co-exist, with only the top level administration directory names being distinct. For example:

Oasys Suite 21.0	**Oasys Suite 21.1**
Top level directory **`OA_ADMIN_21`**	Top level directory **`OA_ADMIN_211`**
oa_pref file in **`OA_ADMIN_21`** contains: **`oasys*install_dir:`** *<pathname for **21.0** installation>* **`oasys*manuals_dir:`** *<pathname for **21.0** manuals>* **`oasys*home_dir:`** *<pathname for home directory>* **`oasys*temp_dir:`** *<pathname for temporary files>*	oa_pref file in **`OA_ADMIN_211`** contains: **`oasys*install_dir:`** *<pathname for **21.1** installation>* **`oasys*manuals_dir:`** *<pathname for **21.1** manuals>* } would almost certainly be unchanged between major } versions, although they could be different if desired
Pathnames in the oa_pref file may contain environment variables which will be resolved before being applied.	

21.1.3. The Hierarchy of oa_pref File Reading

The hierarchy of oa_pref file reading

It will be clear from the above that in a large installation the "oa_pref" files have a significant role. Each piece of software reads them in the following order:

OA_ADMIN_xx	Top level configuration
OA_INSTALL_xx	Installation level
OA_HOME	User's personal "home" file
Current working directory	File specific to the current directory (rarely used)

The rules for reading these files are:

- If a given directory does not exist, or no file is found in that directory, then no action is taken. This is not an error.
- A more recently read definition supersedes one read earlier, therefore "local" definitions can supersede "global" ones (unless it was locked).
- If two of more of the directories in the table above are the same then that file is only read once from the first instance.

21.1.4. Locking Preference Options

Locking Preference Options

From Oasys Suite 9.4 onwards, preference options can be locked. If a preference option is locked in a file then that preference option will be ignored in any of the subsequent preference files that are read.

Therefore by locking a preference in a top-level file in the hierarchy above, eg in **OA_ADMIN_xx**, and then protecting that file to be read-only, an administrator can set preferences that cannot be altered by users since any definitions of that preference in their private oa_pref files will be ignored.

Preferences are locked by using a hash (#) rather than an asterisk (*) between the code name and the preference string. For example:

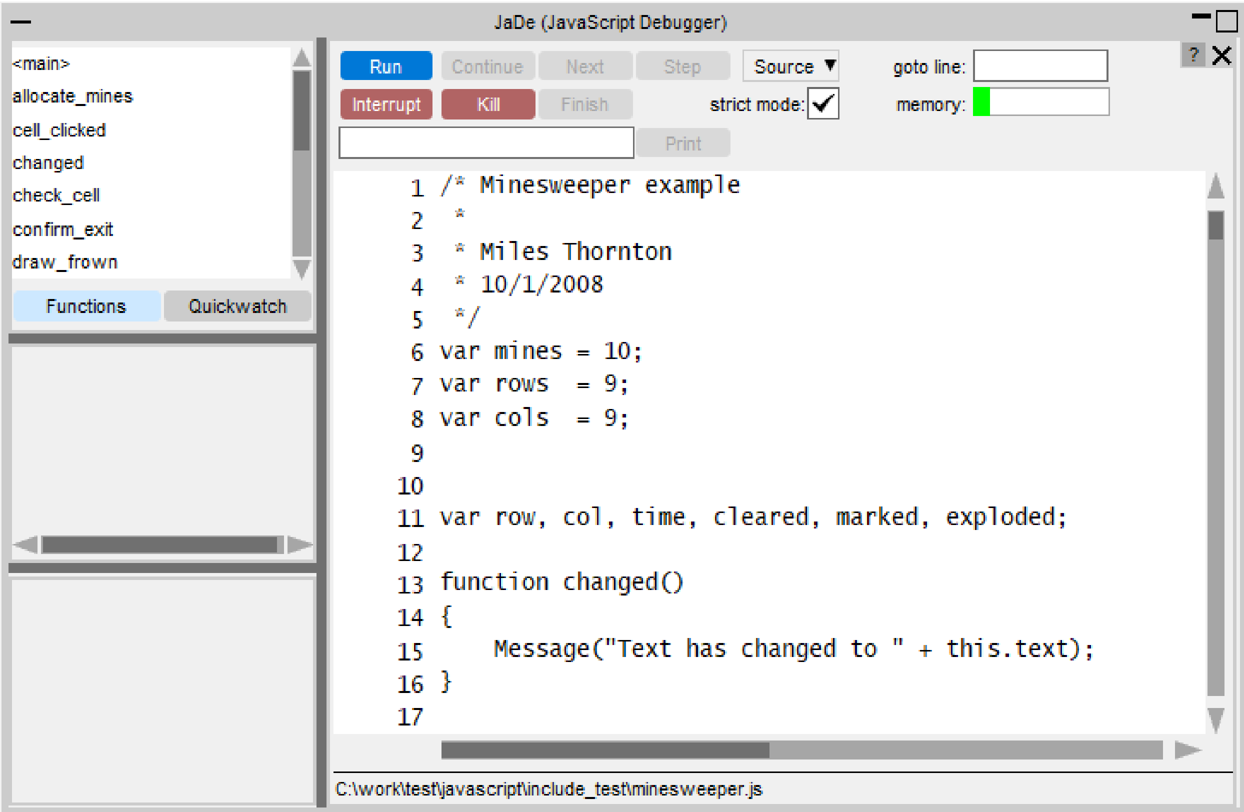
`*maximise: true`	Normal case using "*", means an unlocked preference
`#maximise: true`	Locked case using "#"

These changes may be made either by editing the file manually, or by using the preferences editor.

22. JaDe: The JavaScript debugger

JaDe: The JavaScript debugger

JaDe is included in D3PLOT, PRIMER and T/HIS to help debug and develop JavaScripts. It is started by selecting a script and pressing the **Debug** button in the JavaScript menu in any of the programs. The initial screen is shown below.



It is fairly basic but hopefully has enough functionality for people to be able to find and fix problems in scripts.

22.1. Viewing the Script Files and Functions

Viewing the script files and functions

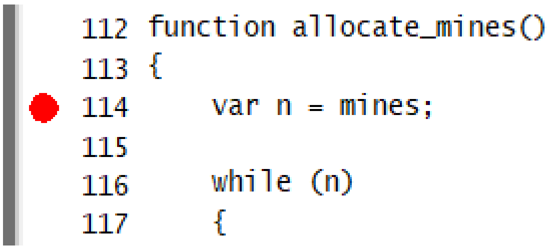
The main part of the window shows the script file. If your script is broken up into separate file (by using Use) then you can get a list of the different files and view them by using the **Source** popup. To go to a particular line in the file use the **goto line** textbox.

A list of the functions in the script is shown in the **Functions** menu on the top left. If you want to look at a particular function then click on the function name and the main text window will jump to the correct file and line.

22.2. Adding/Removing Breakpoints

Adding/removing breakpoints

A breakpoint is a line in the script where execution will pause in JaDe. To add a breakpoint either left click on the line you want the breakpoint on or right click on the line and select **Create breakpoint** from the popup. A red circle is then drawn on the line to show that there is an active breakpoint.

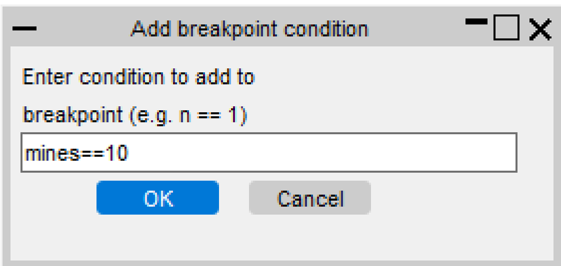


Additionally the breakpoint will also be added to the list in the breakpoint window (bottom left of JaDe). You can click on this at any time and the main text window will jump to the correct file and line.
Active breakpoints are shown with a red circle. Breakpoints can be activated/deactivated by clicking on the line again. Unactive breakpoints are shown as a grey circle instead of a red one. They are also shown in grey text in the breakpoint window .

To delete a breakpoint right click on the line and select **Delete breakpoint**. The breakpoint will be deleted.

Conditional breakpoints

Sometimes it is useful to only stop at a breakpoint if a certain condition is met. For example in the above example we may only want to stop at line 114 if `mines` is 10. You can do this by right clicking on the the breakpoint and selecting **Add condition**.



A window is mapped allowing you type in the condition you want to try to meet. The condition should be a JavaScript expression which evaluates to true if you want the breakpoint to stop execution, or false if you want the breakpoint to be skipped. In this example the condition is `n == 10`.

If a breakpoint has a condition associated with it a C is drawn on the circle and in the breakpoint window. The condition can be edited again or removed by right clicking on the breakpoint and selecting either **Edit condition** or **Remove condition** from the popup.

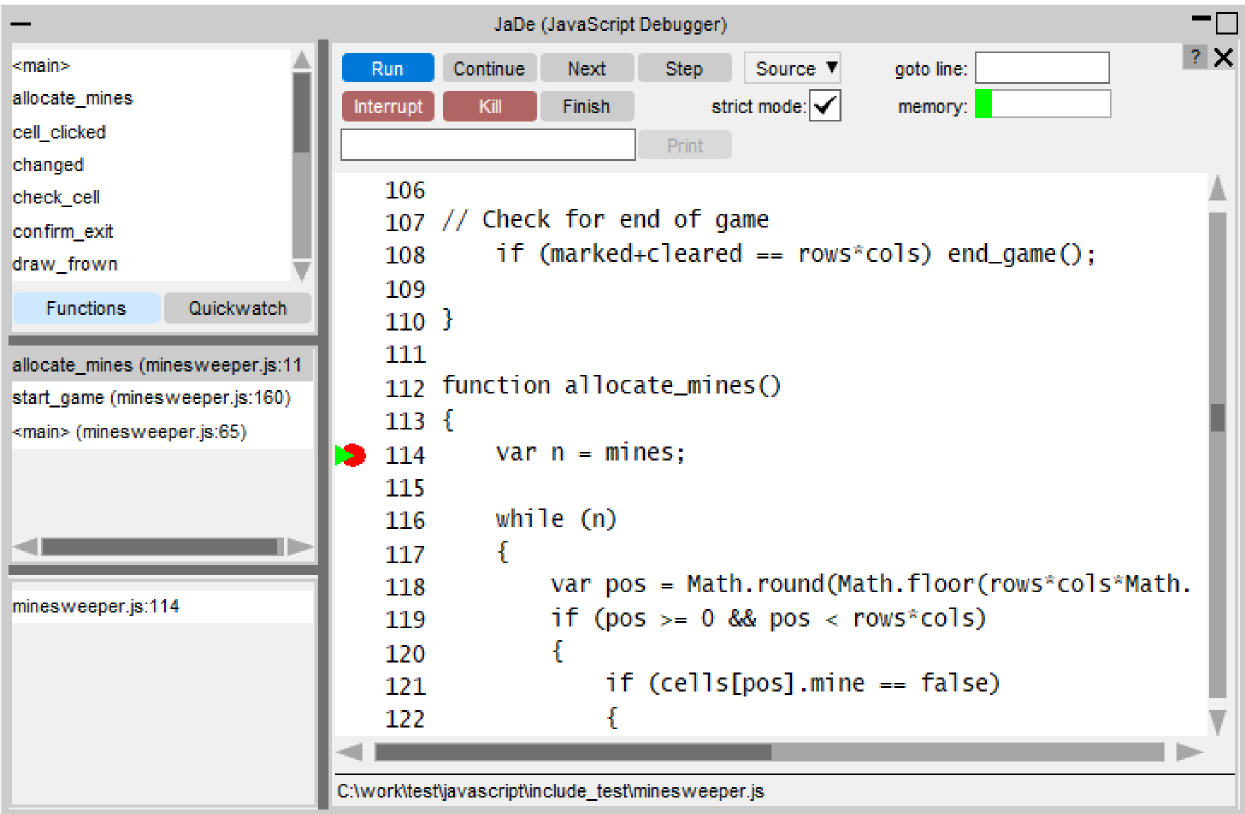
22.3. Running the Script

Running the script

Running the script is controlled by the buttons at the top of the debugger window. By default the script will be run in the debugger in 'strict mode'. This tries to pick up things which you might not have intended by running the script in a stricter environment doing more checking. You can toggle this on/off by using the **strict mode** checkbox.

Starting and stopping

To start the script press the **Run** button. Execution of the script will start. If you have not defined any breakpoints then the script will run until it finishes (unless there are some script errors or exceptions). If there is a breakpoint then the debugger will stop execution of the script when it reaches it. If the script is running and you want to pause execution of the script at any time you can press **Interrupt**.



The line that the debugger has paused the script on is shown by a green triangle. In the above example it is paused at line 114. The middle panel on the left shows the call stack. See the call stack section for more details.

Stepping and continuing

Once the script is paused in the debugger you can step through the source code by using the **Continue**, **Next**, **Step** and **Finish** buttons.

Continue will resume execution of the script again.
Next continues to the next line in the current function. i.e. it will step *over* a function call.
Step continues execution to the next source line (which may be in a different function. i.e. it will step *into* a function call).
Finish will finish executing the current function and stop at the next line in the calling function (the function above this in the call stack).

Alternatively, if you want to continue until a particular line you can right click on the line you want to continue until and select **Continue to here** from the popup.

22.4. Printing the Value of a Variable

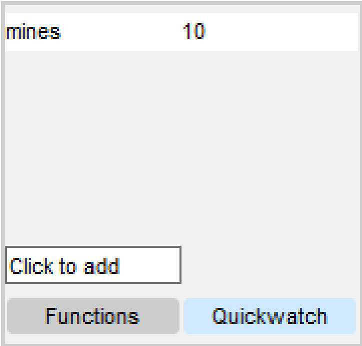
Printing the value of a variable

If you want to see the value of a variable you can type the name of the variable you want to see in the textbox at the top of the debugger and press **Print**. JaDe will evaluate the variable and output the result in the statusbar at the bottom of the debugger.

Using Quickwatch

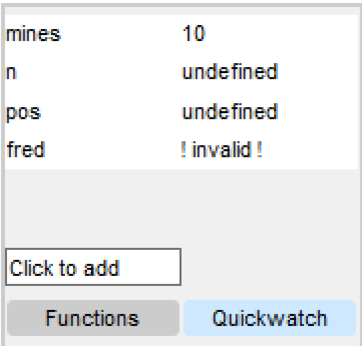
If you want to look at the values for lots of variables it is annoying to have to type the variable name in and press **Print** for each one. A better way is to use **Quickwatch** at the top left of JaDe

Type the name of the variable that you want to watch in the **Click to add** textbox. A line will be added for the variable showing its name and value. e.g. in the following image the variable `mines` is being displayed and its current value is `10`. If the value is very long hover over the value to get the whole string.



You can add any number of variables to watch. To remove one right click on the variable and select **Remove quickwatch** from the popup.

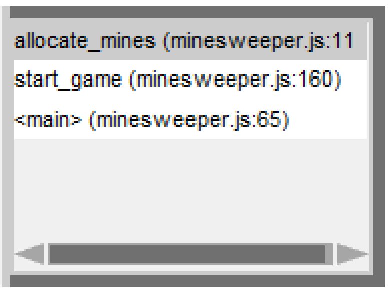
If a variable exists and has been assigned to then the value is displayed. e.g. `mines` in the following example.
If the variable exists but it has not yet had a value assigned its value is the `undefined` value. e.g. `pos` in the following example.
If the variable does not exist the value is shown as `! invalid !`. e.g. `fred` in the following example.



22.5. The Call Stack

The call stack

The call stack shows which functions have been called in the script to get to the current point. It is the middle left window in JaDe.



The top line shows the function that the script is currently paused at. The other lines show the calling functions in order. The above example can be read as:

1. The script starts
2. On line 65 in script file minesweeper.js in the 'main' program the function `start_game` is called.
3. On line 160 in script file minesweeper.js in function `start_game` the function `allocate_mines` is called
4. On line 114 in script file minesweeper.js in function `allocate_mines` the script is paused.

This information is sometimes very useful in more complicated scripts to find out the order things are done in.

The function that the user is currently looking at is highlighted in blue. You can move up or down the call stack by clicking on a line. The main text window will jump to the correct file and line. The line will be shown with a blue triangle instead of a green triangle.

22.6. Exceptions

Exceptions

Sometimes when developing a script you get errors that you need to try to investigate and fix. e.g. an object is null when it should be defined or you try to call a method that does not exist for an object. In these cases an exception is thrown by JavaScript and the script would terminate is run normally. JaDe will trap the exception and stop at the line where the exception occured. e.g. If for example you has the following code:

Copy Code
JavaScript

```
var w = new Window('Example', 0.5, 1.0, 0.5, 1.0);

w.BadMethod();

w.Show()
```

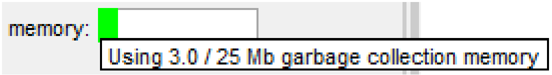
There is no method called `BadMethod` for a `Window`. JaDe will stop at this point and allow you to look at the script.

22.7. Memory Usage

Memory usage

When a script creates arrays, objects or strings it has to allocate some memory to be able to do so (for example an array storing 1,000,000 items will use considerably more memory than an array to store 100 items). To manage this memory JavaScript uses a process called 'garbage collection'. When the array, object or string goes out of scope (can no longer be reached by the script) it can be garbage collected and the memory freed. For the JavaScript engine to be able to do this it must keep track of what memory has been allocated. It does this by keeping a list of the live memory. This list also uses a small amount of memory and this memory is the garbage collection memory. The maximum size for the garbage collection memory is set when running a script.

JaDe allows you do see how much garbage collection memory has been used with a usage bar.



If you hover over the usage bar you can see exactly how much garbage collection memory is being used. As the JavaScript engine allocates memory for objects, arrays etc this will increase. When the engine performs garbage collection to free memory the usage will go down. Note that the engine will normally only perform garbage collection when it thinks it is necessary so if you run a script multiple times in JaDe the memory could continue to increase until the engine decides to do garbage collection, then the memory will reduce.

Note also that JaDe also requires some garbage collection memory to function so the bar also includes some memory for JaDe.

23. Licences Used in Software

Licences used in software

The Oasys LS-DYNA Environment Ltd software uses several third party libraries and executables. The licences for them are given below

23.1. Apple Public Source

Apple Public Source

Note: the URL http://www.apple.com/publicsource cited above no longer exists, see instead https://spdx.org/licenses/APSL-1.1.html

23.2. Draco

Draco

google/draco is licensed under the Apache License:

```
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
See the License for the specific language governing permissions and
limitations under the License.
```

23.3. Expat

Expat

```
Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd
                               and Clark Cooper
Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

Permission is hereby granted, free of charge, to any person
obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject
to
the following conditions:

The above copyright notice and this permission notice shall be
included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE
```

23.4. FreeType

FreeType

Portions of this software are copyright The FreeType Project (www.freetype.org). All rights reserved.

```
                    The FreeType Project LICENSE
                    ----------------------------

                            2006-Jan-27
```

Copyright 1996-2002, 2006 by
David Turner, Robert Wilhelm, and Werner Lemberg

Introduction

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

o We don't promise that this software works. However, we will be interested in any kind of bug reports. (`as is' distribution)

o You can use this software for whatever you want, in parts or full form, without having to pay us. (`royalty-free' usage)

o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (`credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

"""
Portions of this software are copyright <year> The FreeType Project (www.freetype.org). All rights reserved.
"""

Please replace <year> with the value from the FreeType version you actually use.

Legal Terms

```
0. Definitions
--------------

  Throughout this license,  the terms `package', `FreeType Project',
  and  `FreeType  archive' refer  to  the  set  of files  originally
  distributed  by the  authors  (David Turner,  Robert Wilhelm,  and
  Werner Lemberg) as the `FreeType Project', be they named as alpha,
  beta or final release.

  `You' refers to  the licensee, or person using  the project, where
  `using' is a generic term including compiling the project's source
  code as  well as linking it  to form a  `program' or `executable'.
  This  program is  referred to  as  `a program  using the  FreeType
  engine'.

  This  license applies  to all  files distributed  in  the original
  FreeType  Project,   including  all  source   code,  binaries  and
  documentation,  unless  otherwise  stated   in  the  file  in  its
  original, unmodified form as  distributed in the original archive.
  If you are  unsure whether or not a particular  file is covered by
  this license, you must contact us to verify this.

  The FreeType  Project is copyright (C) 1996-2000  by David Turner,
  Robert Wilhelm, and Werner Lemberg.  All rights reserved except as
  specified below.

1. No Warranty
--------------

  THE FREETYPE PROJECT  IS PROVIDED `AS IS' WITHOUT  WARRANTY OF ANY
  KIND, EITHER  EXPRESS OR IMPLIED,  INCLUDING, BUT NOT  LIMITED TO,
  WARRANTIES  OF  MERCHANTABILITY   AND  FITNESS  FOR  A  PARTICULAR
  PURPOSE.  IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS
  BE LIABLE  FOR ANY DAMAGES CAUSED  BY THE USE OR  THE INABILITY TO
  USE, OF THE FREETYPE PROJECT.

2. Redistribution
-----------------

  This  license  grants  a  worldwide, royalty-free,  perpetual  and
  irrevocable right  and license to use,  execute, perform, compile,
  display,  copy,   create  derivative  works   of,  distribute  and
  sublicense the  FreeType Project (in  both source and  object code
  forms)  and  derivative works  thereof  for  any  purpose; and  to
  authorize others  to exercise  some or all  of the  rights granted
  herein, subject to the following conditions:

    o Redistribution of  source code  must retain this  license file
      (`FTL.TXT') unaltered; any  additions, deletions or changes to
      the original  files must be clearly  indicated in accompanying
      documentation.   The  copyright   notices  of  the  unaltered,
      original  files must  be  preserved in  all  copies of  source
      files.
```

```
    o Redistribution in binary form must provide a  disclaimer  that
      states  that  the software is based in part of the work of the
      FreeType Team,  in  the  distribution  documentation.  We also
      encourage you to put an URL to the FreeType web page  in  your
      documentation, though this isn't mandatory.

  These conditions  apply to any  software derived from or  based on
  the FreeType Project,  not just the unmodified files.   If you use
  our work, you  must acknowledge us.  However, no  fee need be paid
  to us.

3. Advertising
--------------

  Neither the  FreeType authors and  contributors nor you  shall use
  the name of the  other for commercial, advertising, or promotional
  purposes without specific prior written permission.

  We suggest,  but do not require, that  you use one or  more of the
  following phrases to refer  to this software in your documentation
  or advertising  materials: `FreeType Project',  `FreeType Engine',
  `FreeType library', or `FreeType Distribution'.

  As  you have  not signed  this license,  you are  not  required to
  accept  it.   However,  as  the FreeType  Project  is  copyrighted
  material, only  this license, or  another one contracted  with the
  authors, grants you  the right to use, distribute,  and modify it.
  Therefore,  by  using,  distributing,  or modifying  the  FreeType
  Project, you indicate that you understand and accept all the terms
  of this license.

4. Contacts
-----------

  There are two mailing lists related to FreeType:

    o freetype@nongnu.org

      Discusses general use and applications of FreeType, as well as
      future and  wanted additions to the  library and distribution.
      If  you are looking  for support,  start in  this list  if you
      haven't found anything to help you in the documentation.

    o freetype-devel@nongnu.org

      Discusses bugs,  as well  as engine internals,  design issues,
      specific licenses, porting, etc.

  Our home page can be found at

    http://www.freetype.org


--- end of FTL.TXT ---
```

23.5. FFmpeg

FFmpeg

```
FFmpeg is licensed under the LGPL v2.1+. The exception to this is
the x264
library used by FFmpeg, for which Arup have obtained a commercial
license (see
here).

# License

Most files in FFmpeg are under the GNU Lesser General Public License
version 2.1
or later (LGPL v2.1+). Read the file `COPYING.LGPLv2.1` for details.
Some other
files have MIT/X11/BSD-style licenses. In combination the LGPL v2.1+
applies to
FFmpeg.

Some optional parts of FFmpeg are licensed under the GNU General
Public License
version 2 or later (GPL v2+). See the file `COPYING.GPLv2` for
details. None of
these parts are used by default, you have to explicitly pass `--
enable-gpl` to
configure to activate them. In this case, FFmpeg's license changes
to GPL v2+.

Specifically, the GPL parts of FFmpeg are:

- libpostproc
- optional x86 optimization in the files
    - `libavcodec/x86/flac_dsp_gpl.asm`
    - `libavcodec/x86/idct_mmx.c`
    - `libavfilter/x86/vf_removegrain.asm`
- the following building and testing tools
    - `compat/solaris/make_sunver.pl`
    - `doc/t2h.pm`
    - `doc/texi2pod.pl`
    - `libswresample/tests/swresample.c`
    - `tests/checkasm/*`
    - `tests/tiny_ssim.c`
- the following filters in libavfilter:
    - `signature_lookup.c`
    - `vf_blackframe.c`
    - `vf_boxblur.c`
    - `vf_colormatrix.c`
    - `vf_cover_rect.c`
    - `vf_cropdetect.c`
    - `vf_delogo.c`
    - `vf_eq.c`
    - `vf_find_rect.c`
```

```
    - `vf_fspp.c`
    - `vf_histeq.c`
    - `vf_hqdn3d.c`
    - `vf_kerndeint.c`
    - `vf_lensfun.c` (GPL version 3 or later)
    - `vf_mcdeint.c`
    - `vf_mpdecimate.c`
    - `vf_nnedi.c`
    - `vf_owdenoise.c`
    - `vf_perspective.c`
    - `vf_phase.c`
    - `vf_pp.c`
    - `vf_pp7.c`
    - `vf_pullup.c`
    - `vf_repeatfields.c`
    - `vf_sab.c`
    - `vf_signature.c`
    - `vf_smartblur.c`
    - `vf_spp.c`
    - `vf_stereo3d.c`
    - `vf_super2xsai.c`
    - `vf_tinterlace.c`
    - `vf_uspp.c`
    - `vf_vaguedenoiser.c`
    - `vsrc_mptestsrc.c`

Should you, for whatever reason, prefer to use version 3 of the
(L)GPL, then
the configure parameter `--enable-version3` will activate this
licensing option
for you. Read the file `COPYING.LGPLv3` or, if you have enabled GPL
parts,
`COPYING.GPLv3` to learn the exact legal terms that apply in this
case.

There are a handful of files under other licensing terms, namely:

* The files `libavcodec/jfdctfst.c`,
`libavcodec/jfdctint_template.c` and
  `libavcodec/jrevdct.c` are taken from libjpeg, see the top of the
files for
  licensing details. Specifically note that you must credit the IJG
in the
  documentation accompanying your program if you only distribute
executables.
  You must also indicate any changes including additions and
deletions to
  those three files in the documentation.
* `tests/reference.pnm` is under the expat license.


## External libraries

FFmpeg can be combined with a number of external libraries, which
sometimes
```

affect the licensing of binaries resulting from the combination.

Compatible libraries

The following libraries are under GPL version 2:
- avisynth
- frei0r
- libcdio
- libdavs2
- librubberband
- libvidstab
- libx264
- libx265
- libxavs
- libxavs2
- libxvid

When combining them with FFmpeg, FFmpeg needs to be licensed as GPL as well by
passing `--enable-gpl` to configure.

The following libraries are under LGPL version 3:
- gmp
- libaribb24
- liblensfun

When combining them with FFmpeg, use the configure option `--enable-version3` to
upgrade FFmpeg to the LGPL v3.

The VMAF, mbedTLS, RK MPI, OpenCORE and VisualOn libraries are under the Apache License
2.0. That license is incompatible with the LGPL v2.1 and the GPL v2, but not with
version 3 of those licenses. So to combine these libraries with FFmpeg, the
license version needs to be upgraded by passing `--enable-version3` to configure.

The smbclient library is under the GPL v3, to combine it with FFmpeg,
the options `--enable-gpl` and `--enable-version3` have to be passed to
configure to upgrade FFmpeg to the GPL v3.

Incompatible libraries

There are certain libraries you can combine with FFmpeg whose licenses are not
compatible with the GPL and/or the LGPL. If you wish to enable these
libraries, even in circumstances that their license may be incompatible, pass
`--enable-nonfree` to configure. This will cause the resulting binary to be
unredistributable.

The Fraunhofer FDK AAC and OpenSSL libraries are under licenses which are
incompatible with the GPLv2 and v3. To the best of our knowledge, they are
compatible with the LGPL.

23.6. HDF5

HDF5

23.7. Jpeg

Jpeg

```
fitness for a particular purpose.  This software is provided "AS
IS", and you,
its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-2012, Thomas G. Lane, Guido
Vollbeding.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute
this
software (or portions thereof) for any purpose, without fee, subject
to these
conditions:
(1) If any part of the source code for this software is distributed,
then this
README file must be included, with this copyright and no-warranty
notice
unaltered; and any additions, deletions, or changes to the original
files
must be clearly indicated in accompanying documentation.
(2) If only executable code is distributed, then the accompanying
documentation must state that "this software is based in part on the
work of
the Independent JPEG Group".
(3) Permission for use of this software is granted only if the user
accepts
full responsibility for any undesirable consequences; the authors
accept
NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the
IJG code,
not just to the unmodified library.  If you use our work, you ought
to
acknowledge us.

Permission is NOT granted for the use of any IJG author's name or
company name
in advertising or publicity relating to this software or products
derived from
it.  This software may be referred to only as "the Independent JPEG
Group's
software".

We specifically permit and encourage the use of this software as the
basis of
commercial products, provided that all warranty or liability claims
are
assumed by the product vendor.
```

23.8. Libcurl

Libcurl

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2012, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.

23.9. Libgif

Libgif

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

23.10. Libpng

Libpng

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE
===

PNG Reference Library License version 2

```
The Contributing Authors and Group 42, Inc. specifically permit,
without fee, and encourage the use of this source code as a
component
to supporting the PNG file format in commercial products.  If you
use
this source code in a product, acknowledgment is not required but
would
be appreciated.
```

23.11. Libxlsxwriter

Libxlsxwriter

Libxlsxwriter includes `queue.h` from FreeBSD and the `minizip`
component of
`zlib` which have the following licenses:

Queue.h from FreeBSD:

23.12. libzip

libzip

libzip is released under a 3-clause BSD license:

23.13. MPEG-LA

MPEG-LA

23.14. Openssl

Openssl

```
LICENSE ISSUES
  ==============

  The OpenSSL toolkit stays under a double license, i.e. both the
conditions of
  the OpenSSL License and the original SSLeay license apply to the
toolkit.
  See below for the actual license texts.

  OpenSSL License
  ---------------

/*
====================================================================
 * Copyright (c) 1998-2017 The OpenSSL Project.  All rights
reserved.
 *
 * Redistribution and use in source and binary forms, with or
without
 * modification, are permitted provided that the following
conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above
copyright
 *    notice, this list of conditions and the following disclaimer
in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL
Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be
used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please
contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called
"OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
```

```
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the
following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL
Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND
ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 *
====================================================================
 *
 * This product includes cryptographic software written by Eric
Young
 * (eay@cryptsoft.com).  This product includes software written by
Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

 Original SSLeay License
 -----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes
SSL.
 *
 * This library is free for commercial and non-commercial use as
long as
 * the following conditions are aheared to.  The following
conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL
documentation
```

```
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available
version or
 * derivative of this code cannot be changed.  i.e. this code cannot
simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

23.15. PCRE2

PCRE2

```
PCRE2 LICENCE
-------------

PCRE2 is a library of functions to support regular expressions whose
syntax
and semantics are as close as possible to those of the Perl 5
language.

Releases 10.00 and above of PCRE2 are distributed under the terms of
the "BSD"
licence, as specified below, with one exemption for certain binary
redistributions. The documentation for PCRE2, supplied in the "doc"
directory,
is distributed under the same terms as the software itself. The data
in the
testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding.
Also
included in the distribution is a just-in-time compiler that can be
used to
optimize pattern matching. This is an optional feature that can be
omitted when
the library is built.


THE BASIC LIBRARY FUNCTIONS
---------------------------

Written by:       Philip Hazel
Email local part: ph10
Email domain:     cam.ac.uk
```

University of Cambridge Computing Service,
Cambridge, England.

PCRE2 JUST-IN-TIME COMPILATION SUPPORT

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

STACK-LESS JUST-IN-TIME COMPILER

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

THE "BSD" LICENCE

23.16. PDFHummus

PDFHummus

Is licensed under the Apache License:

23.17. POV-Ray

POV-Ray

Is licensed under the GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007 which may be found here http://www.povray.org/povlegal.html

Oasys Ltd use the POV-Ray executable in unmodified form as a separate, stand-alone entity. We have not modified the source code or the executable in any way.

We convey the executable as part of our installation package, and in accordance with the licence:

- Users who install POV-Ray must accept the licence terms cited above.
- We provide a download of the POV-Ray executable and source code on our website http://www.oasys-software.com/dyna/en/

23.18. Schemasafe

23.19. SmoothSort

SmoothSort

Oasys Ltd acknowledge Wikibooks as the source of this algorithm, which is used in unmodified form.

23.20. Spidermonkey

Spidermonkey

```
Mozilla Public License Version 2.0
==================================

1. Definitions
--------------

1.1. "Contributor"
    means each individual or legal entity that creates, contributes
to
    the creation of, or owns Covered Software.

1.2. "Contributor Version"
    means the combination of the Contributions of others (if any)
used
    by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
    means Covered Software of a particular Contributor.

1.4. "Covered Software"
    means Source Code Form to which the initial Contributor has
attached
    the notice in Exhibit A, the Executable Form of such Source Code
    Form, and Modifications of such Source Code Form, in each case
    including portions thereof.

1.5. "Incompatible With Secondary Licenses"
    means

    (a) that the initial Contributor has attached the notice
described
        in Exhibit B to the Covered Software; or

    (b) that the Covered Software was made available under the terms
of
        version 1.1 or earlier of the License, but not also under
the
        terms of a Secondary License.

1.6. "Executable Form"
    means any form of the work other than Source Code Form.

1.7. "Larger Work"
    means a work that combines Covered Software with other material,
in
    a separate file or files, that is not Covered Software.

1.8. "License"
    means this document.
```

```
1.9. "Licensable"
    means having the right to grant, to the maximum extent possible,
    whether at the time of the initial grant or subsequently, any
and
    all of the rights conveyed by this License.

1.10. "Modifications"
    means any of the following:

    (a) any file in Source Code Form that results from an addition
to,
        deletion from, or modification of the contents of Covered
        Software; or

    (b) any new file in Source Code Form that contains any Covered
        Software.

1.11. "Patent Claims" of a Contributor
    means any patent claim(s), including without limitation, method,
    process, and apparatus claims, in any patent Licensable by such
    Contributor that would be infringed, but for the grant of the
    License, by the making, using, selling, offering for sale,
having
    made, import, or transfer of either its Contributions or its
    Contributor Version.

1.12. "Secondary License"
    means either the GNU General Public License, Version 2.0, the
GNU
    Lesser General Public License, Version 2.1, the GNU Affero
General
    Public License, Version 3.0, or any later versions of those
    licenses.

1.13. "Source Code Form"
    means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
    means an individual or a legal entity exercising rights under
this
    License. For legal entities, "You" includes any entity that
    controls, is controlled by, or is under common control with You.
For
    purposes of this definition, "control" means (a) the power,
direct
    or indirect, to cause the direction or management of such
entity,
    whether by contract or otherwise, or (b) ownership of more than
    fifty percent (50%) of the outstanding shares or beneficial
    ownership of such entity.

2. License Grants and Conditions
--------------------------------
```

```
2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free,
non-exclusive license:

(a) under intellectual property rights (other than patent or
trademark)
    Licensable by such Contributor to use, reproduce, make
available,
    modify, display, perform, distribute, and otherwise exploit its
    Contributions, either on an unmodified basis, with
Modifications, or
    as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell,
offer
    for sale, have made, import, and otherwise transfer either its
    Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution
become effective for each Contribution on the date the Contributor
first
distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted
under
this License. No additional rights or licenses will be implied from
the
distribution or licensing of Covered Software under this License.
Notwithstanding Section 2.1(b) above, no patent license is granted
by a
Contributor:

(a) for any code that a Contributor has removed from Covered
Software;
    or

(b) for infringements caused by: (i) Your and any other third
party's
    modifications of Covered Software, or (ii) the combination of
its
    Contributions with other software (except as part of its
Contributor
    Version); or

(c) under Patent Claims infringed by Covered Software in the absence
of
    its Contributions.

This License does not grant any rights in the trademarks, service
marks,
```

```
or logos of any Contributor (except as may be necessary to comply
with
the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to
distribute the Covered Software under a subsequent version of this
License (see Section 10.2) or under the terms of a Secondary License
(if
permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its
Contributions are its original creation(s) or it has sufficient
rights
to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under
applicable copyright doctrines of fair use, fair dealing, or other
equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses
granted
in Section 2.1.

3. Responsibilities
-------------------

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including
any
Modifications that You create or to which You contribute, must be
under
the terms of this License. You must inform recipients that the
Source
Code Form of the Covered Software is governed by the terms of this
License, and how they can obtain a copy of this License. You may not
attempt to alter or restrict the recipients' rights in the Source
Code
Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code
    Form, as described in Section 3.1, and You must inform
recipients of
```

```
    the Executable Form how they can obtain a copy of such Source
Code
    Form by reasonable means in a timely manner, at a charge no more
    than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this
    License, or sublicense it under different terms, provided that
the
    license for the Executable Form does not attempt to limit or
alter
    the recipients' rights in the Source Code Form under this
License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your
choice,
provided that You also comply with the requirements of this License
for
the Covered Software. If the Larger Work is a combination of Covered
Software with a work governed by one or more Secondary Licenses, and
the
Covered Software is not Incompatible With Secondary Licenses, this
License permits You to additionally distribute such Covered Software
under the terms of such Secondary License(s), so that the recipient
of
the Larger Work may, at their option, further distribute the Covered
Software under the terms of either this License or such Secondary
License(s).

3.4. Notices

You may not remove or alter the substance of any license notices
(including copyright notices, patent notices, disclaimers of
warranty,
or limitations of liability) contained within the Source Code Form
of
the Covered Software, except that You may alter any license notices
to
the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support,
indemnity or liability obligations to one or more recipients of
Covered
Software. However, You may do so only on Your own behalf, and not on
behalf of any Contributor. You must make it absolutely clear that
any
such warranty, support, indemnity, or liability obligation is
offered by
You alone, and You hereby agree to indemnify every Contributor for
any
liability incurred by such Contributor as a result of warranty,
support,
```

indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all
end user license agreements (excluding distributors and resellers) which
have been validly granted by You or Your distributors under this License
prior to termination shall survive termination.

****
*
*
* 6. Disclaimer of Warranty
*
* -------------------------
*
*
*
* Covered Software is provided under this License on an "as is"
*
* basis, without warranty of any kind, either expressed, implied, or *
* statutory, including, without limitation, warranties that the
*
* Covered Software is free of defects, merchantable, fit for a
*
* particular purpose or non-infringing. The entire risk as to the
*
* quality and performance of the Covered Software is with You.
*
* Should any Covered Software prove defective in any respect, You
*
* (not any Contributor) assume the cost of any necessary servicing,
*
* repair, or correction. This disclaimer of warranty constitutes an
*
* essential part of this License. No use of any Covered Software is
*
* authorized under this License except under this disclaimer.
*
*
*
****

****
*
*
* 7. Limitation of Liability
*
* --------------------------
*

```
*
*
*  Under no circumstances and under no legal theory, whether tort
*
*  (including negligence), contract, or otherwise, shall any
*
*  Contributor, or anyone who distributes Covered Software as
*
*  permitted above, be liable to You for any direct, indirect,
*
*  special, incidental, or consequential damages of any character
*
*  including, without limitation, damages for lost profits, loss of
*
*  goodwill, work stoppage, computer failure or malfunction, or any
*
*  and all other commercial damages or losses, even if such party
*
*  shall have been informed of the possibility of such damages. This
*
*  limitation of liability shall not apply to liability for death or
*
*  personal injury resulting from such party's negligence to the
*
*  extent applicable law prohibits such limitation. Some
*
*  jurisdictions do not allow the exclusion or limitation of
*
*  incidental or consequential damages, so this exclusion and
*
*  limitation may not apply to You.
*
*
*
**********************************************************************
****

8. Litigation
-------------

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of
that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous
----------------

This License represents the complete agreement concerning the
subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
```

necessary to make it enforceable. Any law or regulation which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section
10.3, no one other than the license steward has the right to modify or
publish new versions of this License. Each version will be given a
distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version
of the License under which You originally received the Covered Software,
or under the terms of any subsequent version published by the license
steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a
modified version of this License if you rename the license and remove
any references to the name of the license steward (except to note that
such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary
Licenses

If You choose to distribute Source Code Form that is Incompatible With
Secondary Licenses under the terms of this version of the License, the
notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public
License, v. 2.0. If a copy of the MPL was not distributed with this

```
  file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular
file, then You may include the notice in a location (such as a
LICENSE
file in a relevant directory) where a recipient would be likely to
look
for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice
---------------------------------------------------------

  This Source Code Form is "Incompatible With Secondary Licenses",
as
  defined by the Mozilla Public License, v. 2.0.
```

23.21. TOML Parser for C

TOML Parser for C

23.22. Treeview

Treeview

23.23. Turf

Turf

23.24. Win-iconv

Win-iconv

23.25. x264

x264

The x264 software library is used under commercial license from x264, LLC

23.26. Zlib

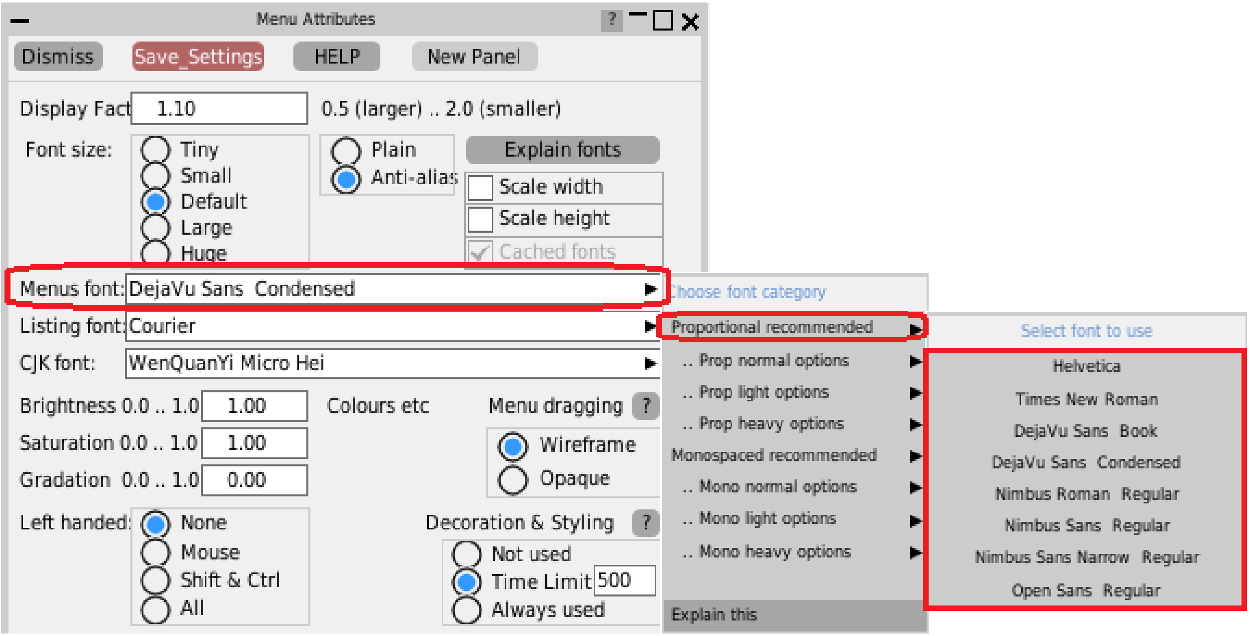
Zlib

24. Fonts on Linux

Fonts on Linux

Prior to Oasys Suite 17.0 the Oasys Ltd. LS-DYNA environment software used "legacy" X11 fixed fonts on Linux, from Oasys Suite 17.0 onwards, the software uses Freetype fonts, which give improved appearance and a wider range of typefaces.

The recommended proportional font for menu panels is "DejaVu Sans Condensed" which is widely available on Linux, but you can change this using **Options, Menu attributes** where a different font can be selected from those available on your system. For example on the author's CentOS 7 system the choice of fonts is:



24.1. The Range of Fonts Available

The range of fonts available

The range of fonts you see on your system will depend on the version of Linux you are using and what fonts you have installed; the image above was captured from a CentOS 7 machine.

The Oasys software interrogates the font server to extract all available fonts, then sorts them for presentation purposes by spacing (proportional or monospaced) and weight (normal, light, bold). The "recommended" fonts, as shown in the right hand popup menu above, are simply those which have been found by trial and error to give the best appearance. However this is a very subjective matter, and you may prefer something different: choose something that you like then use **Save Settings** to save it. If you change your mind later you can always come back to this panel to select something else.

Helvetica is provided as an option for backwards compatibility with the older user interface; it is not natively available on Linux so a different font is substituted, which tends not to look very good in Freetype.

Monospaced font selection problems

We have observed that while proportional font selection works correctly on Linux, the selection of monospaced fonts seems to have some bugs:

- The default "courier" font works, but tends to produce a font that is too small in some situations and probably is not exactly courier, although it looks very similar.
- The "recommended" monospaced font on some systems comes out as "Courier 10 Pt Regular", which is a genuine courier font, however if you select that it will produce something completely different. Experiment shows that if you ask for "Courier 10 Pt" then you get what you expect, but appending "Regular" breaks the font selection somehow

This appears to be a "fontconfig" problem: the system's font server simply gets it wrong. This can be demonstrated by the command

`fc-match` "*font of your choice*"

for example **`fc-match "courier"`** on a RHEL 7 machine produces the result **`"Nimbus Mono PS" "Regular"`**

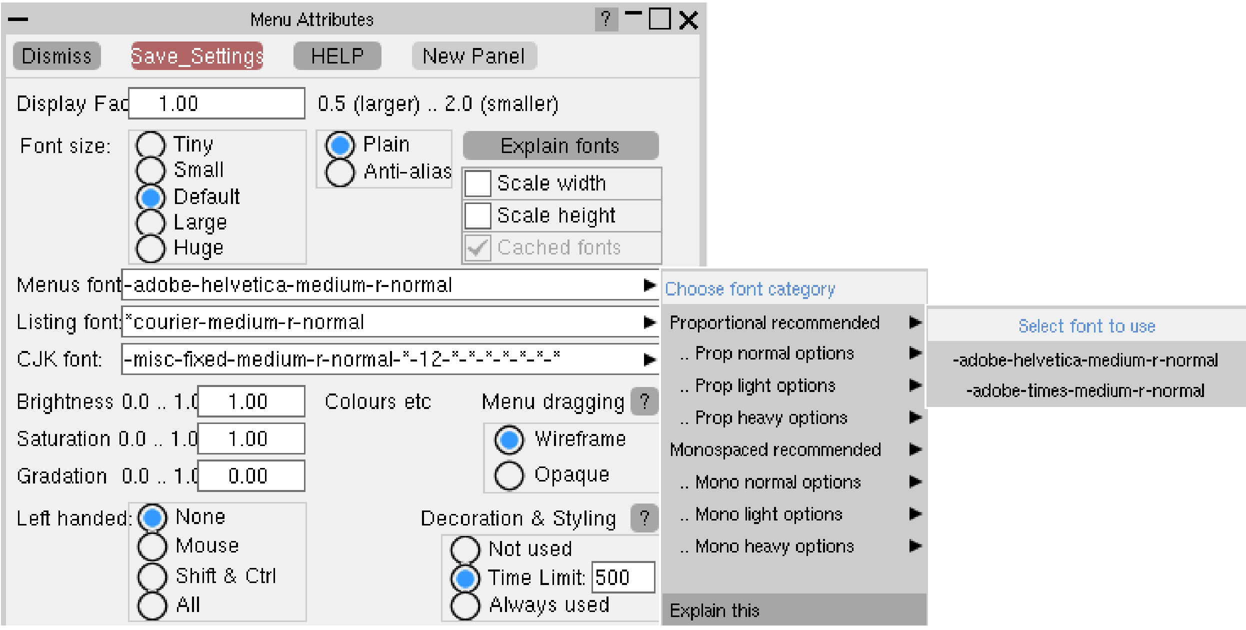
If you are happy with the monospaced font used for help texts and the like you don't need to take any action, however if you want to change it you may need to experiment a bit to find something that looks good on your system by typing different variations of names into the "Listing font: [.....]" text entry box. You can use the "fc-match" command in conjunction with this to see what the font server will map your request onto. Once you have found something satisfactory use **Save Settings** to save it in your oa_pref file and it will be remembered for future use.

24.2. Plain Versus Anti-aliased Fonts

Plain versus Anti-aliased fonts

On some monitors, especially relatively low resolution ones, the anti-aliasing of fonts can result in quite fuzzy text. The quality of this will depend on the version of Freetype installed, and more recent Linuxes will tend to look better since they are more likely to use sub-pixel sampling.

Some users may prefer the cruder but sharper appearance of the original "core X11" legacy fonts, and these can be used by changing to **Plain** so long as you actually have these fonts loaded on your machine. On the CentOS 7 machine being used to create this manual page the equivalent "plain" font image of the above is:



If you try this on your machine and it doesn't work then it means that you need to load the legacy font package(s), see below.

Loading legacy Core X11 fonts

You don't need to load these, it is only necessary if you want the old-style "plain" appearance described in the section above.

You will need root privileges to install these, so unless you are familiar with working as root and using commands such as "rpm", "yum" or "yast" please seek help from your IT department, or alternatively contact Oasys Ltd for help.

The best fonts to install are the 75 dots per inch (dpi) ones, which can be obtained online for a range of common Linux operating systems from https://pkgs.org/download/xorg-x11-fonts-75dpi

If that fails you may already have the relevant packages in your installation files, you should look for (in order)

RedHat/CentOS

```
xorg-x11-fonts-75dpi
xorg-x11-fonts-ISO8859-1-75dpi
xorg-x11-fonts-Type1
xorg-x11-fonts-misc
```

xorg-x11-fonts-100dpi
xorg-x11-fonts-ISO8859-1-100dpi

You don't have to install all of these.

The 75dpi and 100dpi font packages are the same typefaces at different resolutions. You should choose the one which gives the best looking results on your display, but in the author's experience the 75dpi one looks fine but the 100dpi one looks as if a spider was let loose with a leaky pen! Always try the 75dpi one first.

To manage fonts on RHEL/CentOS do the following:

- Log in as root
- To see the X11 fonts currently installed type "**`yum list installed | grep xorg | grep font`**"
- To see X11 fonts available but not installed "**`yum list available | grep xorg | grep font`**"
- To install something "**`yum install`** *package*", for example "**`yum install xorg-x11-fonts-75dpi`**"

You can list the range of "yum" commands available with "man yum".

SUSE

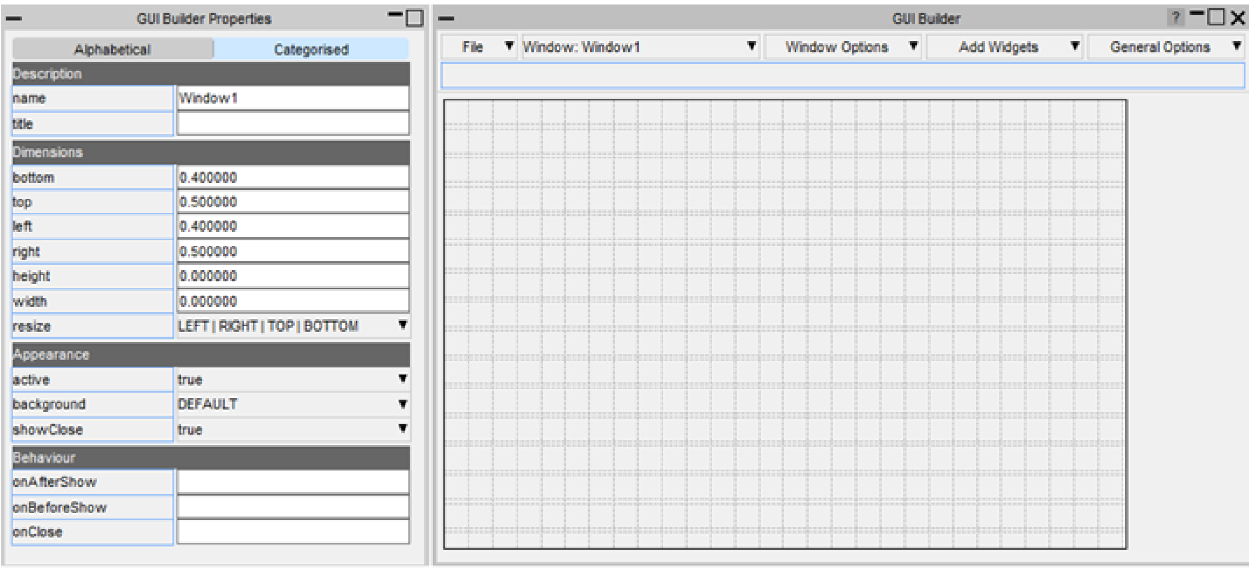
xorg-x11-fonts-core
xorg-x11-fonts

25. The JavaScript GUI Builder

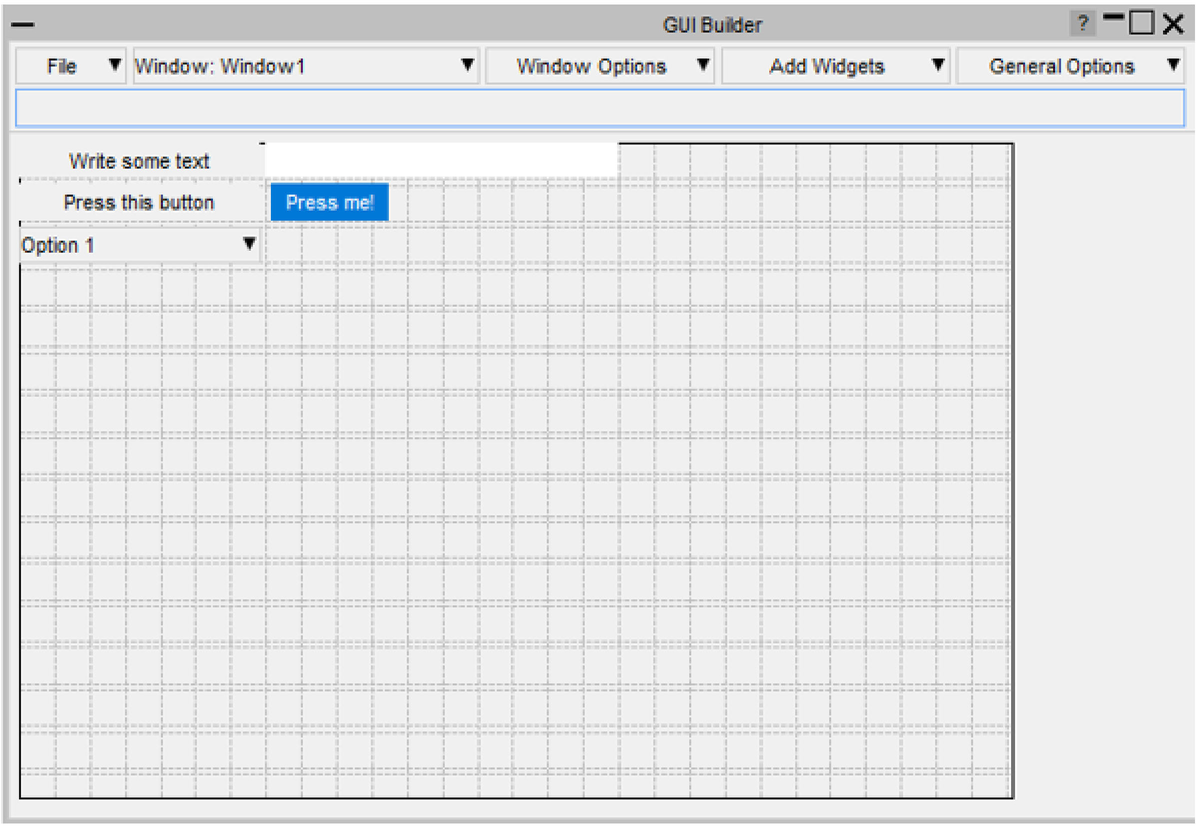
The JavaScript GUI Builder

The JavaScript GUI Builder is an interactive GUI Builder, available in D3PLOT, PRIMER and T/HIS, making it easier to create JavaScript GUIs, removing the need to write code to create windows and widgets.

It can be started by pressing the **GUI Builder** button in the JavaScript menu in any of the programs.



You can then design and save your GUI to a file:

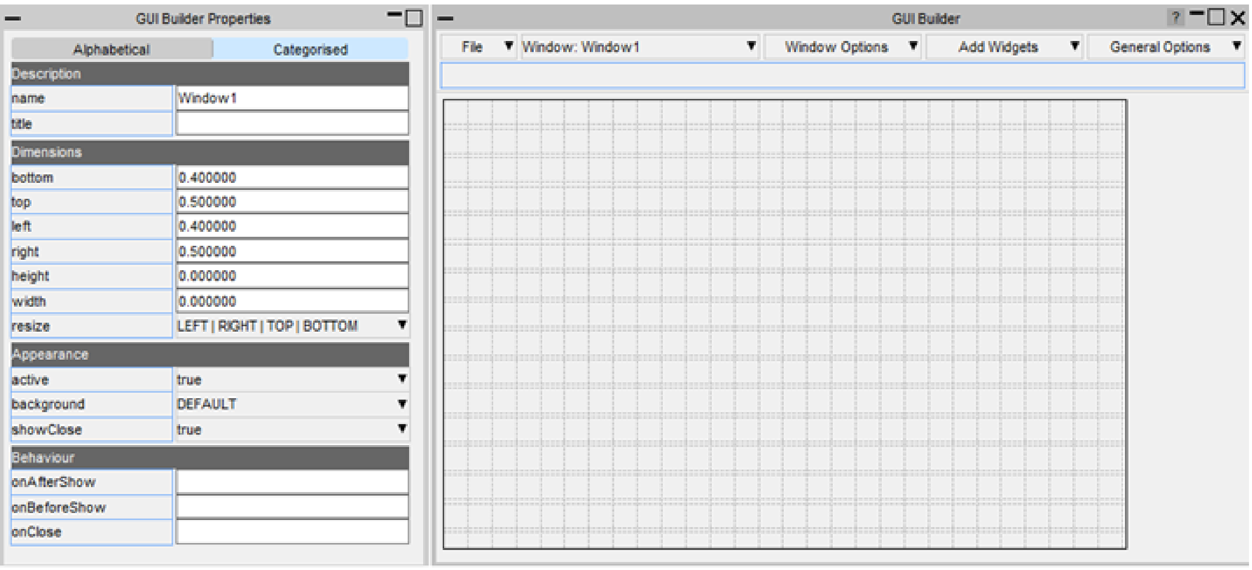


Then read the file in your script to automatically generate the window and widgets:

25.1. How to Build a GUI

How to build a GUI

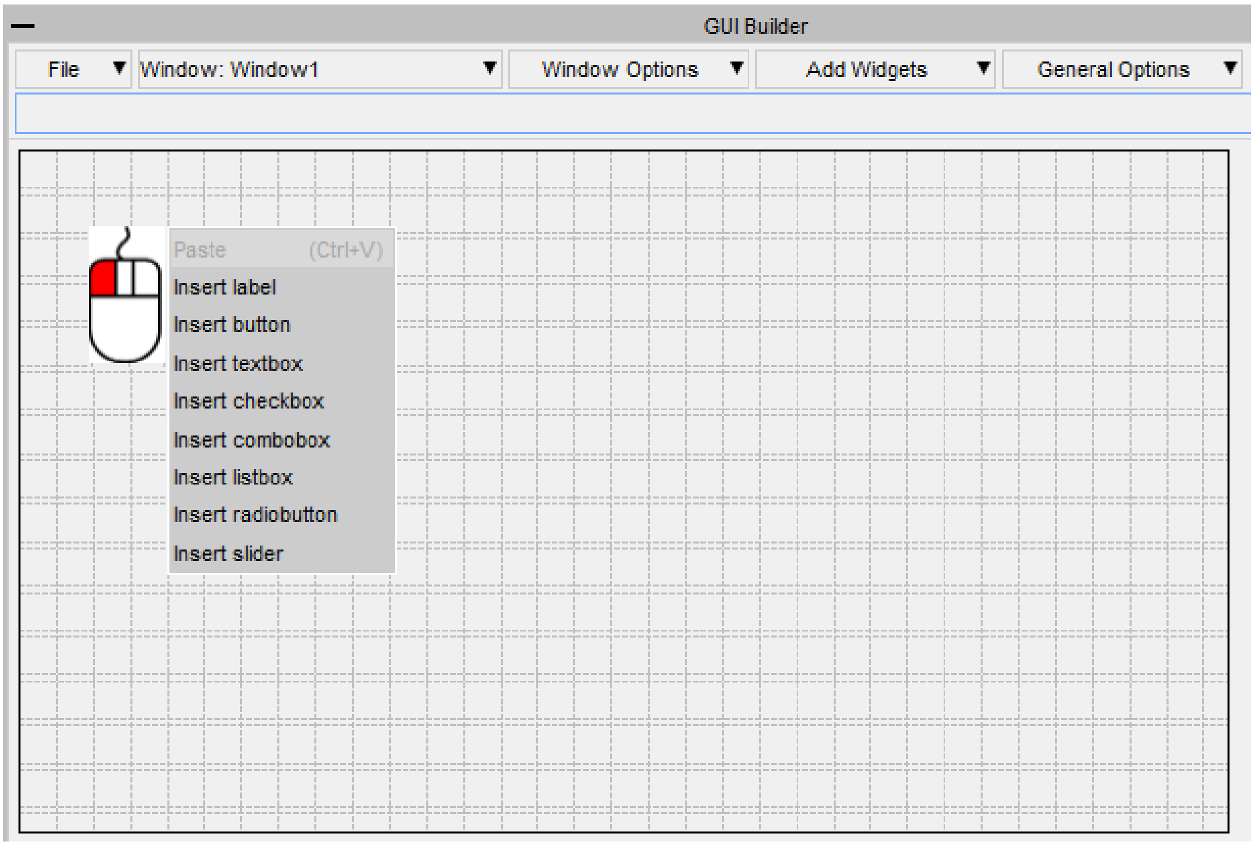
The builder is split into two windows. The properties window for setting the properties of the widgets and windows and a design window for adding, positioning and resizing widgets.

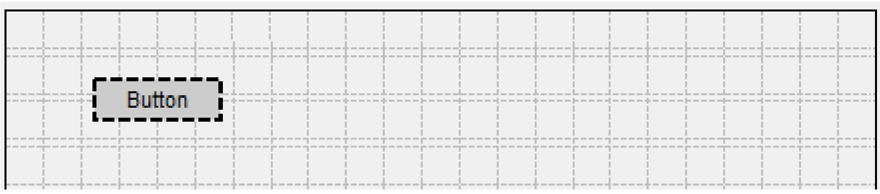


25.1.1. Add a Widget

Add a widget

Widgets can be added by right-clicking on the design window and selecting the widget type to add. The widget will be added with default properties and highlighted with dashed lines to indicate that it's the current widget.

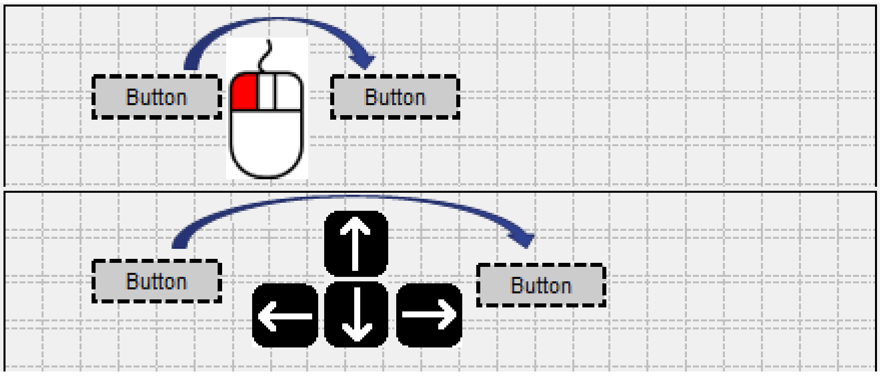




25.1.2. Move a Widget

Move a widget

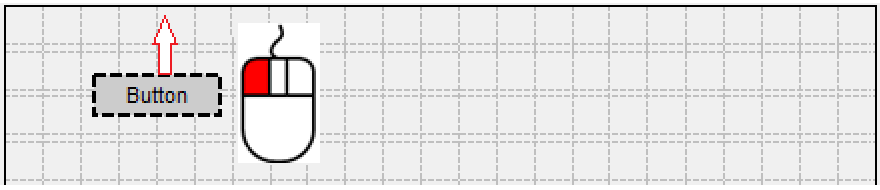
Widgets can be moved by left-clicking on them and dragging, or by using arrow keys.

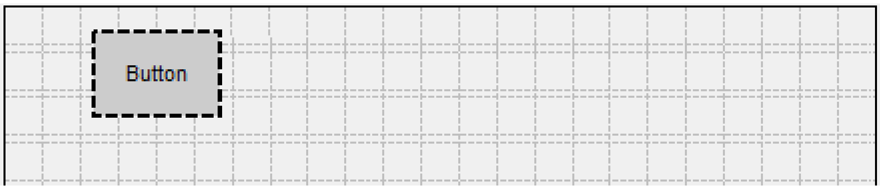


25.1.3. Resize a Widget

Resize a widget

Widgets can be resized by left-clicking on their border and dragging.

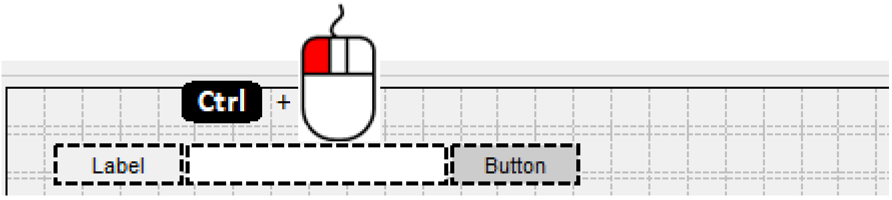




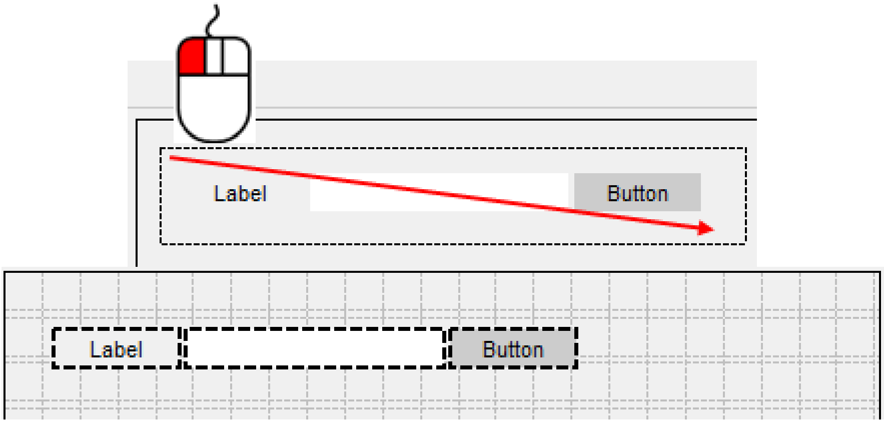
25.1.4. Selecting Widgets

Selecting widgets

Multiple widgets can be selected by holding the Ctrl or Shift keys and left-clicking.



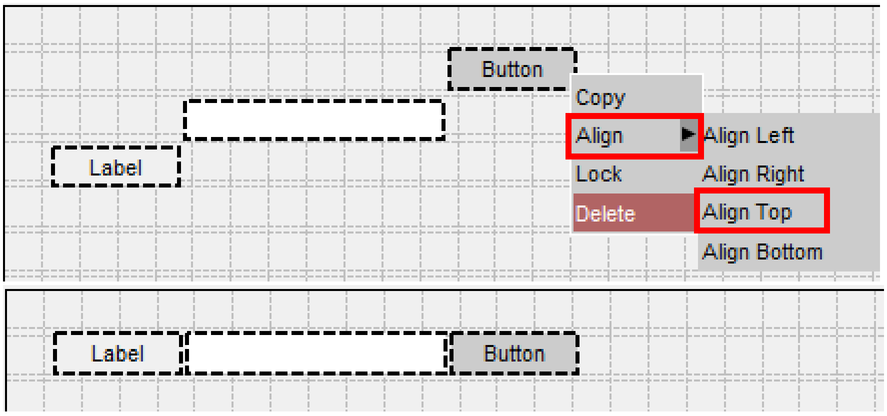
Alternatively a box can be dragged around the widgets you want to select.



25.1.5. Aligning Widgets

Aligning widgets

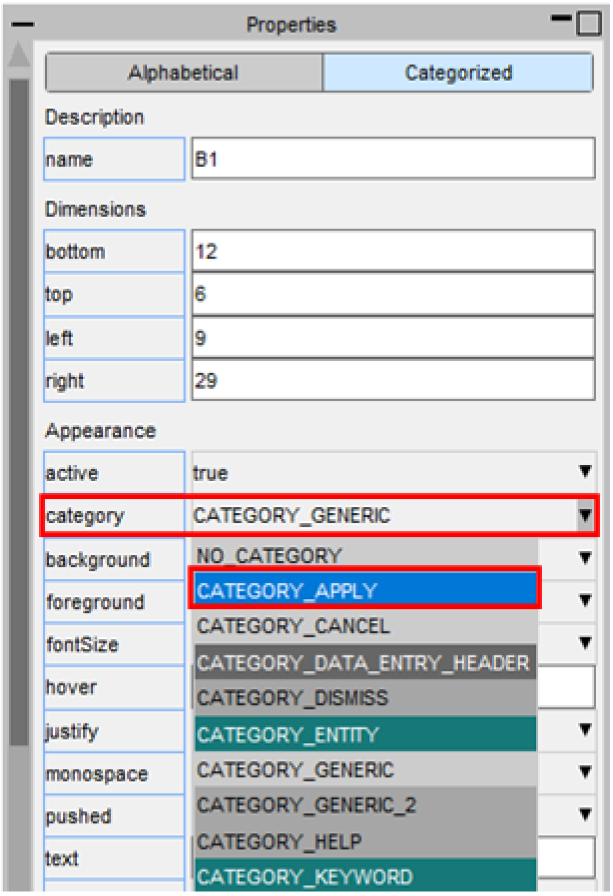
When multiple widgets are selected the borders can be aligned by right-clicking on the widget you want to align the other widgets to, and then selecting how you want them to be aligned.



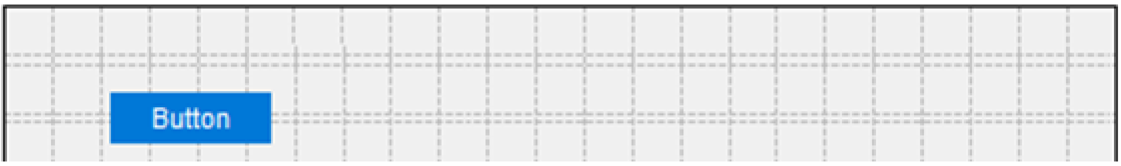
25.1.6. Setting the Properties of Widgets

Setting the properties of widgets

The properties of a widget can be modified in the properties window, e.g. change the category to CATEGORY_APPLY.



The appearance of the widget will update in the design window. If multiple widgets are selected the property will be applied to all the selected widgets.

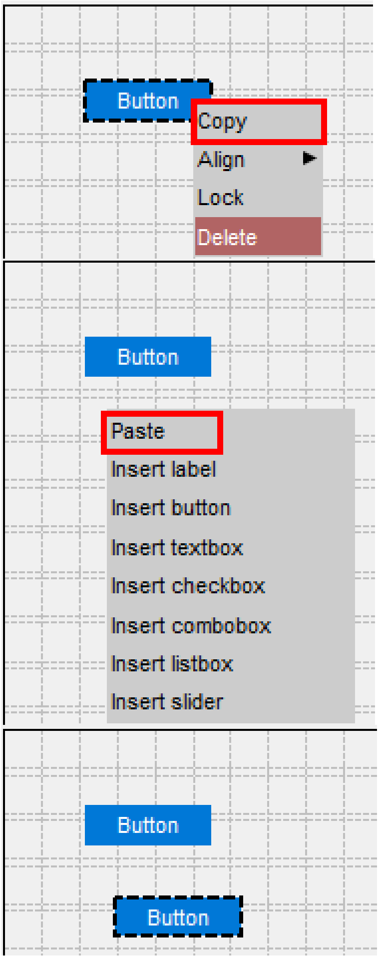


25.1.7. Copying and Pasting Widgets

Copying and pasting widgets

You can copy and paste widgets by right-clicking on them and selecting **Copy** and then right-clicking on the window and selecting **Paste**. The new widget will have all the same properties as the copied widget.

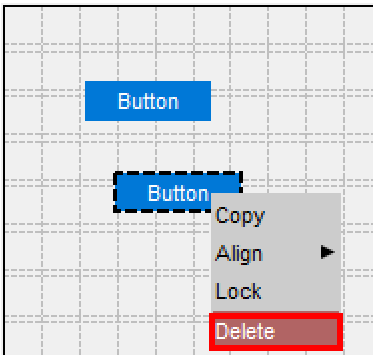
Alternatively you can use the shortcuts Ctrl-C and Ctrl-V.



25.1.8. Deleting Widgets

Deleting widgets

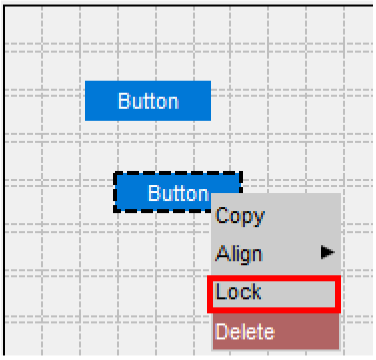
To delete a widget, right-click on it and select **Delete**. Alternatively you can press the Delete shortcut key.



25.1.9. Lock the Position of Widgets

Lock the position of widgets

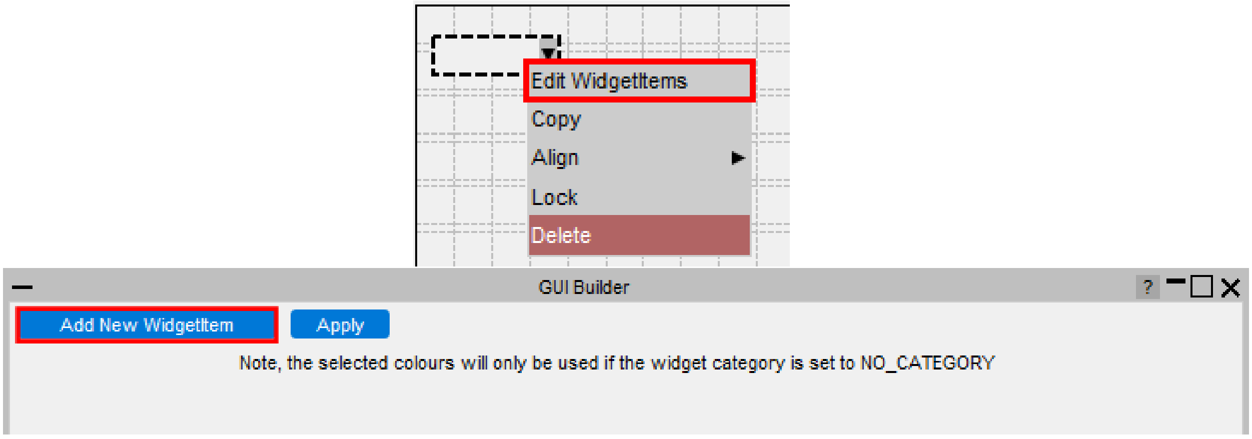
To lock the position of a widget so it can't be repositioned or resized, right-click on it and select **Lock**. To unlock it again, right-click on it and select **Unlock**



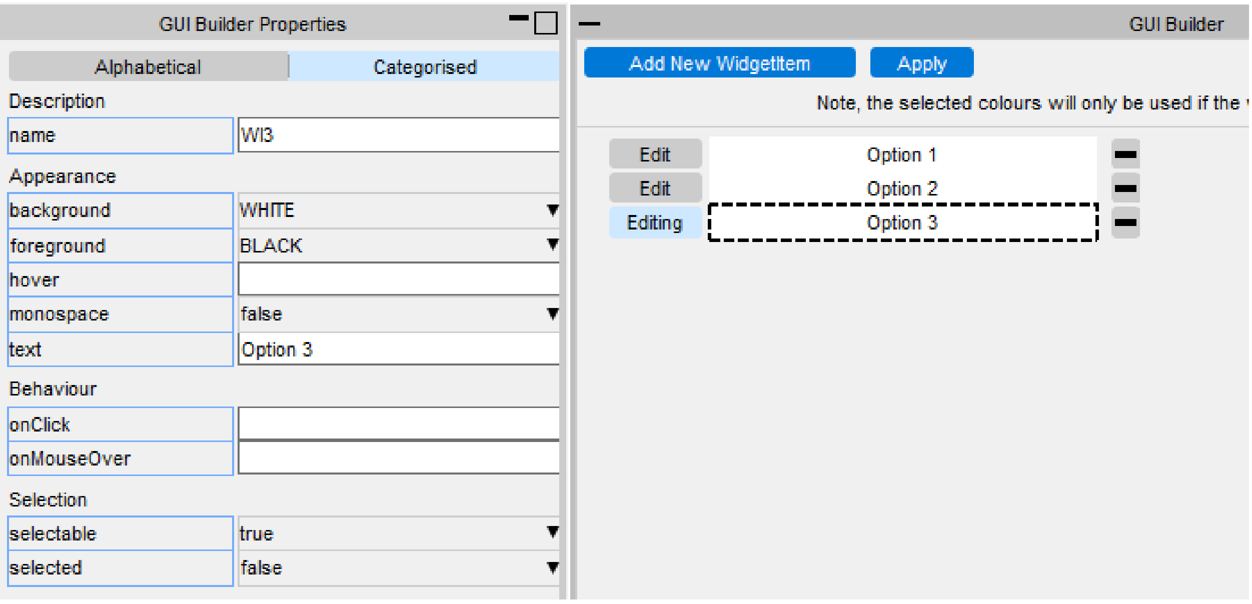
25.1.10. Adding Widgetitems to Comboboxes and Listboxes

Adding widgetitems to comboboxes, listboxes and radiobuttons

To add WidgetItems to a Combobox, Listbox or Radiobutton, right-click on it and select **Edit WidgetItems**. This will update the design window where you can add WidgetItems by pressing the **Add New WidgetItem** button.



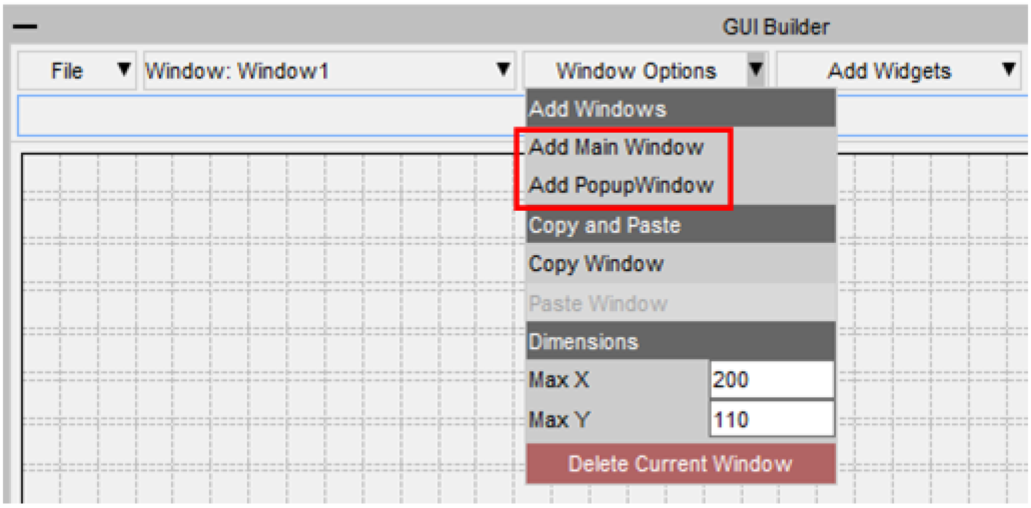
The appearance of the current WidgetItem can be modified in the same way as Widgets by clicking on the WidgetItem and updating its properties. To delete a WidgetItem, click on the **-** on the right hand side. Once you have finished, press **Apply** to return to the normal design window.



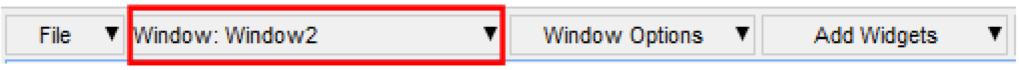
25.1.11. Adding Windows

Adding windows

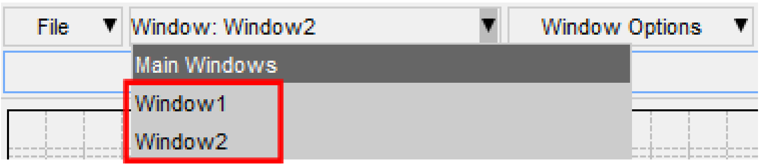
Additional windows can be created by clicking on the Window Options dropdown menu. You can add either a Main Window or PopupWindow.



The name of the current window is displayed in the Window selection dropdown menu.



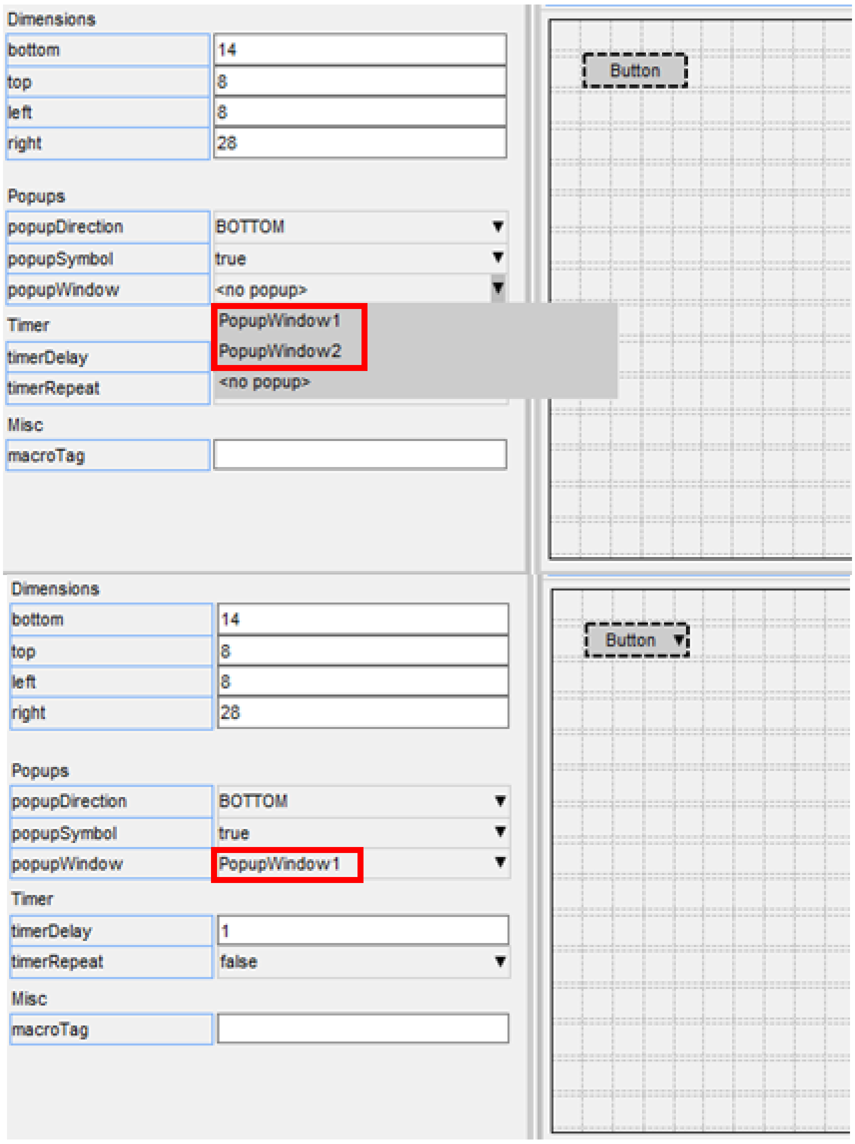
To change to a different window, select it from the dropdown menu.



25.1.12. PopupWindows

PopupWindows

PopupWindows can be linked to widgets by setting the popupWindow property.

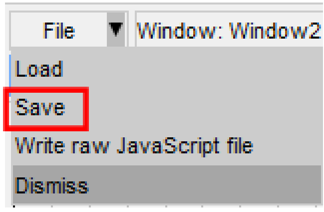


To remove a PopupWindow linked to a widget, set the popupWindow to <no popup>.

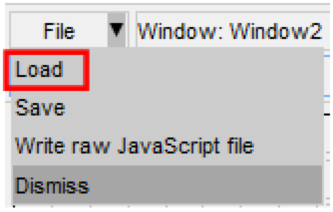
25.1.13. Saving and Loading a GUI

Saving and loading a GUI

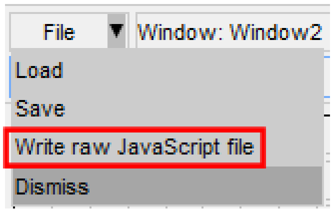
The GUI can be saved to file by pressing the **Save** button and then selecting a file. The saved file is a JavaScript file containing the window and widget definitions in a JSON string, and a call to Window.BuildGUIFromString() which builds the GUI when the script is run. Further details are given in the next section.



It can be reloaded by pressing the **Load** button and selecting the file to load.



The GUI can also be saved as a raw JavaScript file, with the calls to create and position the windows and widgets, explicitly defined, rather than using Window.BuildGUIFromString(). This cannot be loaded back into the GUI Builder, however it may be useful for creating GUIs to run in versions prior to v18 that don't have the Window.BuildGUIFromString() function.



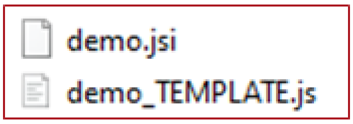
25.2. How to Use the GUI in a Script

How to use the GUI in a script

The GUI is saved to a JavaScript file, containing the GUI definition in a JSON string and a call to Window.BuildGUIFromString(). It is saved with the extension `.jsi` to indicate that it should be included from another file. You should not need to edit this file.

When saving the GUI a *.js file is also written to demonstrate how to include the *.jsi file and display the GUI. This can be used as a template to follow and modify.

It is written to the same folder as the *.jsi file and named **<jsi_filename>_TEMPLATE.js**, e.g. if the *.jsi file is called **`demo.jsi`**, the *.js file will be saved as **`demo_TEMPLATE.js`**



The following sections explain how you can reference the Windows, Widgets and WidgetItem objects within your script.

25.2.1. Read the GUI Into a Script

Read the GUI Into a Script

To read the GUI in a script you need to include the *.jsi file with the Use() function.

This will create a global variable (`gui` by default) containing all the GUI objects. The name of the variable can be changed in the GUI builder menu under General Options.

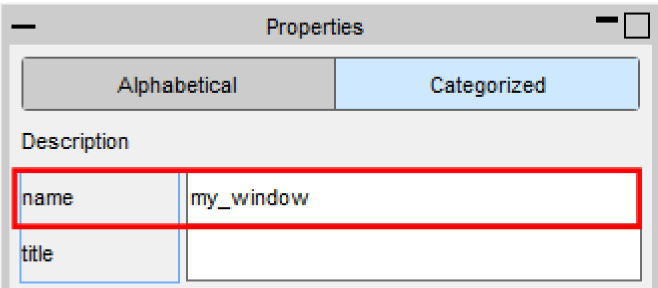
For example, to include the GUI saved in C:\my_gui.jsi:

Use("C:\\my_gui.jsi");

25.2.2. Accessing the Window Objects

Accessing the Window objects

The GUI Window objects are stored as properties on the global GUI object. The name of the property is whatever was defined in the properties window in the GUI builder.



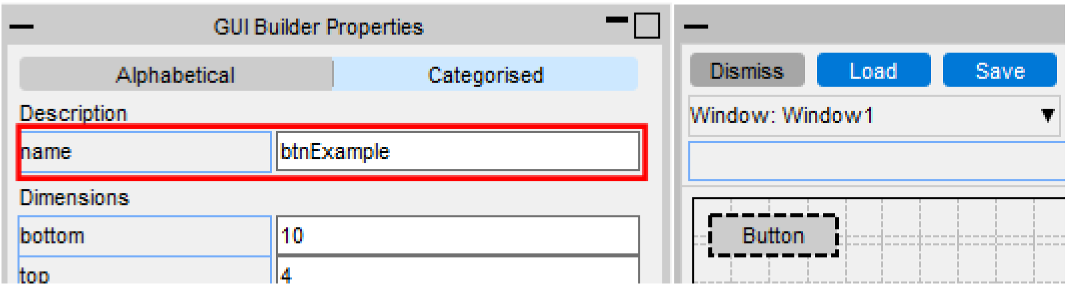
To display the Window called **my_window** use the Show() method:

if (gui) gui.my_window.Show();

25.2.3. Accessing the Widget Objects

Accessing the Widget objects

Similarly, each Widget object is a property of the Window object. The name of the Widget property is whatever was defined in the properties window in the GUI builder.



For example if the window is called **my_window** and the widget is called **btnExample**, the Widget object can be accessed and modified with.

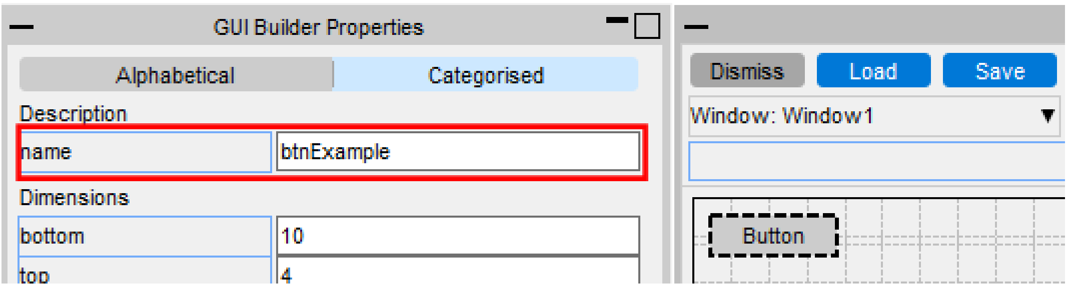
var btn = gui.my_window.btnExample;

btn.text = Test;

25.2.4. Accessing the Widgetitem Objects

Accessing the WidgetItem objects

WidgetItem objects are a property of the Widget.



For, example if the Window is called **my_window**, the Widget the WidgetItem is on is called **cbxExample** and the widget item is called **wi1**, it can be accessed and modified with.

var wi = gui.my_window.cbxExample.wi1;

25.2.5. Defining Callback Functions

Defining callback functions

Callback functions (onClick, onChange, etc.) can be assigned to the window and widgets in the properties window, by adding the name of a function to call.

For example to set the onClick property of a widget so it calls a function called **pressed**:

Functions	
onClick	pressed
onPopup	
onTimer	

This function then needs to be defined in your script:

Use("C:\\test.jsi");

if (gui) gui.my_window.Show();

function pressed()

{

Message("You clicked me!");

}