Oasys Ltd Software Suite

Version 12.0

Update and Release Notes

Oasys Ltd

The Software house of Arup

Contents

			Page
1	**Introduction**		**1**
	1.1	Compatibility with LS-DYNA releases	1
	1.2	Hardware platforms supported	1
	Graphics hardware supported		1
	1.3	FLEXlm Licensing	1
2	**Bugs Fixed**		**3**
	2.1	PRIMER	3
	2.2	D3PLOT	39
	2.3	T/HIS	47
	2.4	REPORTER	51
	2.5	Shell	52
3	**Enhancements**		**53**
	3.1	PRIMER	53
	3.2	D3PLOT	75
	3.3	T/HIS	80
	3.4	REPORTER	87

1 Introduction

The Oasys Ltd LS-DYNA Environment 12.0 suite, dated October 2014 is a full release which supersedes all previous releases.

This document lists the enhancements that have been made during development, and the bugs that have been fixed (since 11.1). Each “case id” referred to in these lists is unique and should be cited if more information or clarification are required.

1.1 Compatibility with LS-DYNA releases

The 12.0 release has been used and tested with LS-DYNA version R7.1, and supports all the keywords in volumes I, II and III of the R7.1 User’s Manual dated May 2014

LS-DYNA R7.1 is now the default keyword output format from PRIMER.

1.2 Hardware platforms supported

The complete Oasys Ltd LS-DYNA Environment 12.0 is available on the following platforms:

Windows	Vista, Windows 7, Windows 8/8.1	32 bit
	Vista 64, Windows 7 64bit, Windows 8/8.1 64bit	32 and 64 bit
Linux	Redhat Enterprise 4.x Redhat Enterprise 5.x Redhat Enterprise 6.x SUSE Enterprise 11.0	32 and 64 bit

For simplicity of support and maintenance only the RedHat Enterprise 5.x build has been released on the website as this has been found to run on most machines.

If you require a different Linux build, please contact us.

Graphics hardware supported

The default graphics protocol used on all platforms in the table above is OpenGL, which gives good quality 3D performance on a wide range of hardware.

“Legacy” X-Windows (2D only) graphics drivers remain in PRIMER and D3PLOT on Linux platforms since these are used by a small minority of clients who display over a network of mixed machine types.

1.3 FLEXlm Licensing

The version 12.0 software has been compiled using version 11.11.0 of FLEXlm (the same as the version 11.0 and 11.1 releases)

The version 11.0 licence file and licence daemons are backwards compatible with both the 10.x and 9.x software releases. This means that any existing 10.x and 9.x software will continue to work with the new licence files and servers.

Note :

1) On Windows the version of "lmgrd.exe" that is installed with the version 12 software been changed to 11.11.1. Version 11.11.1 is supported on Windows Server 2012 while the 11.11.0 release shipped with the version 11 software isn't.

2) We recommend that you update all of your Oasys Ltd LS-DYNA environment license servers with the files shipped with the version 12.0 software.

2 Bugs Fixed

The enhancements and bug-fixes for each program have been broken down into a series of topics. Within each topic enhancements and bug-fixes are listed by case ID (most recent first).

2.1 PRIMER

2.1.1 Bugs Fixed in 12

Airbag

- **Case 25804**

It was possible to corrupt airbag reference geometry data when animating *BOUNDARY_PRESCRIBED_MOTION within PRIMER AND displaying the reference geometry. This has been corrected.

- **Case 24836**

When reading a model containing MAT34 definitions that were not used by any elements PRIMER would show a dialogue box asking the user to choose an option. This was confusing and so has been replaced with a default action and a message for each material.

- **Case 24474**

PRIMER could crash when visualising *INITIAL_AIRBAG_PARTICLE_POSITION. This has been corrected.

- **Case 24472**

PRIMER would fail to read a model containing an airbag particle card if the card referenced individual chambers (CHM_ID) but did not reference an owner chamber card (CHM).

This is an error in the model, but it is better to continue reading and issue an error rather than just failing to read. PRIMER now does this.

- **Case 23749**

PRIMER could crash when creating chambers when daisy chaining from an airbag particle card you are also creating. This has been corrected.

- **Case 22264**

The *DEFINE_CPM_CHAMBERS individual chambers edit panel was restrictive in how to add part sets. This has been corrected.

Assembly

- **Case 24458**

Reading a part tree assembly and the JavaScript model methods Import() and ImportInclude() overwrote the model filename and path. This has now been fixed.

Attached

- **Case 23513**

The Attached.SetEntity() JavaScript function did not work for contacts (i.e. you could not turn on/off the 'find through' flag for contacts in JS). This has been corrected.

Also the Attached.SetEntity() function could reset all switches to OFF when called for the first time. This has been corrected.

Belts

- **Case 24533**

The seatbelt Auto-Refit process got a bit confused if asked to refit an existing belt that:

- was made from a mixture of 1d belt elements and shells.

- did not have a retractor at its start on the B-Post

It would tend to treat the initial stretch from B-Post anchorage to shoulder slipring as having mixed shell / 1d belt elements, rather than using pure 1d belt elements.

This was because the 'work out what is in existing belt' algorithms did not consider this arrangement, and got a bit confused.

Now fixed: end points that are not rettractors but nevertheless use all 1d belt elements to the next slipring are now treated in the same way during auto refit.

- **Case 24515**

The (seatbelt) New Dummy capability in V11 did not work very well, was difficult to understand, and was of only marginal usefulness.

It has been completely rewritten in V12 and is much improved:

- The panel layout has been revised to make functionality clearer, and in particular better visual feedback of the current status and what still needs to be done has been much improved.

- Various internal bugs have been fixed, making the whole process much more robust.

This capability should now be genuinely useful, and should greatly simplify the problem of replacing a dummy in a belted situation.

- **Case 24512**

An example model in which the labels of beams and 1d seatbelt elements overlap was provided. Prior to output PRIMER detects this and warns that Dyna's default of putting beams on springs and 1d seatbelt elements, using the same labels, means that this clash is not allowed and offers to renumber things.

If this renumbering option is used then 1d belt elements are renumbered correctly to avoid a clash, but the and seatbelt elements in the belt fitter *BELT_MESH cards now no longer encompass all 1d belt elements in that seatbelt definition. This doesn't matter until the deck is written out, reread and the belt refitted since the write out / reread cycle recalculates which 1d belt elements are in the belt definition from the first/last labels, and this is no longer correct. The

refit fails to find enough label space in its label 'slot' because some of the 1d belt elements no longer 'belong' to it.

The problem arises because of the vulnerability of using [first] and [last] labels to describe a range of 1d belt elements, which is an expedient solution to the problem that there is no *SET_SEATBELT keyword in LS-DYNA. This implicitly assumes that the belt elements will occupy a contiguous range of labels within those limits and, crucially, also assumes that 1d belt element labels will not be moved from inside that range to outside it. In this instance some were relabelled to lie outside that range, leading to their being 'lost' by the belt fitter following the keyword deck write/reread sequence.

This problem has been much reduced, if not totally solved, by adding a scan for what the first and last labels of a seatbelt definition actually are immediately prior to writing it out to the keyword file. This means that any relabelling that may have taken place will be accounted for, and the first / last labels will be truly representative of what is in the belt.

It is not a total solution because a sufficiently perverse relabelling exercise could result in some elements not originally in the belt being accidentally now included. However this is fairly unlikely as the automatic relabelling built into PRIMER will attempt to minimise label changes, and it would require a fairly perverse user intent on major changes to trigger this problem.

This logic is applied to all definitions under the post-END *BELT cards that use first/last syntax: 1d seatbelt elements, sliprings, retractors, database cross-sections and node sets used for nodal rigid bodies.

- **Case 23947**

*ELEMENT_SEATBELT_SLIPRING is normally bi-directional, with the belt able to pull through in either direction. However it is possible to set a DIRECT flag to limit this to 1->2, or 2->1.

In the case of a 2D slipring using a non-zero DIRECT value it seems that the location of EDGSET (on *SECTION_SHELL) for that stretch of 2d belt elements influences the values reported for pull-through at the slipring, and to give the correct results EDGSET should be located on the 'downstream' (pulled through to) side.

Unfortunately this can conflict with the requirement to locate EDGSET at a retractor if that is in this section of belt, so this is probably a bug in LS_DYNA and has been reported to LSTC. Pending resolution of this PRIMER adopts the following logic:

- If this stretch of 2d belt elements contains a retractor then EDGSET is located there.

- Otherwise if the stretch contains a slipring EDGSET is located at the downstream side as determined by DIRECT.

This logic may need to change when LSTC fix this bug, but the above is correct for PRIMER 12 development as of Feb 2014.

- **Case 23352**

The EDGSET definition on the *SECTION_SHELL card determines the layout of a stretch of 2d belt elements, and if this set is empty or refers to nodes / elements that are not on a 2d belt LS-DYNA terminates with an error in the input phase.

If the seatbelt fitter is used to remesh an existing 2d belt into a different type, ie 1d belt elements and/or shells, it will leave the unused *SET_SHELL definition behind, although it should

remove any EDGSET definition on this card. It has been reported that EDGSET gets left behind in this situation, although it has not proved possible to replicate this problem.

However an extra function has been added to the checking of *SET_SHELL that will detect an 'orphan' EDGSET definition that does not refer to 2d belt elements, giving the option of removing this.

- **Case 21957**

There is a limit to the labels that may be used on *ELEMENT_SEATBELT_SLIPRING definitions due to formatting limitations in LS-DYNA.

For a 1D slipring:

Neither node nor seatbelt elements may have a label > 99,999,999

For a 2D slipring:

Node set may not have a label < -9,999,999

Shells sets may not have labels < -99,999,999

Previously PRIMER did not trap these limits, but they have now been added to the standard

- **Case 20073**

Two ergonomic problems related to seatbelt fitting have been fixed:

(1) In the 'Fit' panel you can make changes to the belt path in a mini editor, but these changes were 'scratch' and not saved in the belt definition unless you did an 'Accept'. Since you had to 'Save' in the editor this was a bit counter-intuitive, and a user could reasonably expect that the 'save' would be enough to make his changes permanent.

Therefore the internal logic has changed so that this is now the case: changes to the belt path become permanent as soon as you 'Save' them.

(2) When moving belt points by dragging it is sometimes useful to swap temporarily to 'Quick pick' mode (eg using short cut key 'q') in order to blank things that are in the way.

This works, but it moves the 'ownership' of the cursor picking process from the belt fitting window to the main graphics one, and in order to revert to dragging points it is necessary to tell the code that you want to go back to this mode. Clicking on the [+] symbol in the top left corner of the seatbelt fitting box, or repeating the 'q' short cut key, would achieve this; however some users also assumed that by clicking again on the appropriate belt path editor button such as 'modify coords' this too would return picking ownership to the seatbelt panel. The latter didn't work because the editor thought 'I'm still in modify coordinates mode', and therefore that no action was required to re-acquire ownership of cursor picking.

This has been fixed by making sure that clicking on any of the mouse-driven belt editor buttons (Delete points, Modify Coords, Control Twist) will always re-acquire ownership of picking if necessary. (This change went into V11.1 as well as V12)

Blanking

- **Case 24660**

IGES Curves might fail to be selectable for screen-picking, either explicitly or by area selection, and this might appear to occur in a rather random fashion.

This was caused by internal errors when marking these curves as 'visible' when drawn, and has been fixed.

Case

- **Case 26154**

If a model contained more than approximately 50 child cases then PRIMER could generate errors during keyword input, and not process the deck correctly.

The error was due to populating an internal matrix of case combinations with NxN (where N = number of child cases) case combination entries, even though the vast majority of these would never actually occur. This overflowed internal storage space and caused the error.

This has been fixed by only creating internal combinations of 2 or more child cases, where a combination is a nested CASE_BEGIN/END definition, when they actually occur.

Checking

- **Case 25724**

Custom checks did not work for materials if there were different material types in the model (which is very likely). Now fixed.

- **Case 25676**

Model check may incorrectly report error that MAT169 solids do not align with shell planes when THKDIR setting is zero. Now fixed.

- **Case 25050**

Problems have arisen with the joint coincidence check in LS-DYNA which requires that pairs of nodes on joint types 1 - 7 are within 1.0e-3 of each other. This check is not related to the size or units of the model, and is an absolute value.

PRIMER checks for this and will identify any nodal pairs at joints which breach this limit, but this check has occasionally gone wrong in two ways:

(1) PRIMER has found the nodal separation to be < 1.0e-3, so acceptable, but LS-DYNA finds it to be less and thus generates an error.

(2) Passage of an input deck through PRIMER can cause subtle small changes to nodal coordinates that cause nodes to separate very slightly, once again triggering this check in LS-DYNA.

The problem seems to arise because of the single precision storage used for nodal coordinates in PRIMER, which limits the precision of numbers to just over 7 decimal places.

This is fine when a node is close to the origin, but when it is moved some distance away the magnitude of its coordinate gets larger, and the precision required to compute the 1.0e-3 absolute value becomes more doubtful.

Testing reveals that for coordinates between 8192.0 and 16384.0 the problem is critical: values below 8192.0 retain sufficient precision, and values above 16384.0 lose the 4th decimal place

entirely, however within that range the 4th decimal place is essentially noise, and this can cause errors in the calculation.

To fix this two changes have been made:

(1) The existing nodal coincidence check for joints in PRIMER examines the distance of the joint from the origin, and if it is further than 8192.0 units away it uses a test of 0.5e-3 rather than 1.0e-3.

This takes into account errors in the precision with which numbers are stored, and makes it slightly more likely that an error will be flagged and the coordinates of the nodes merged to an average value, but a difference of 0.5 in 10,000,000 will make no practical difference to a model.

(2) Prior to keyword output there is a second, silent check for joint nodal coincidence.

This considers each [x,y,z] axis of nodal pairs at joints separately, and merges the coordinates in a given axis if:

- Their magnitude is greater than 8192.0

- They are less than 1.5e-3 apart.

The intention of this check is to remove any spurious errors that may arise because of small changes in precision introduced by PRIMER, but to do so in a way that minimises changes to the model.

- **Case 24611**

An existing error check that throws a warning if some but not all nodes on any shell have a reference geometry definition has been modified. This warning will now only be issued if said shell is actually used in an airbag.

- **Case 24598**

Contact Automatic General containing only seatbelt elements, incorrectly gave message that slave side was empty. Now fixed.

- **Case 24324**

Autofix to remove empty set from *DATABASE_HISTORY_NODE,etc could cause crash. Now fixed.

- **Case 23883**

The error message issued when solid element formulation 13 is applied to a part of material type that does not support it has been clarified. In these cases, the element formulation is silently converted to type 10 which can cause modelling errors.

- **Case 23882**

When is defined for discrete beam of non-zero length, the node is now tested for alignment with N1-N2 and and error message given if this is the case.

- **Case 23452**

The autofix which enforces coincidence for joint nodes did not work well in the case of a node shared by multiple joints. Fixing one joint would break another. The logic has been upgraded to average the position of all the nodes concerned.

Additionally, an out of colinear cylindrical joint (warning) will only be fixed where the nodes are not shared with other joints.

Compare

- **Case 25422**

In Model Modified it is possible to sketch changed items from the report tree, and this was being slow for the case of modified PARTs in some models if many parts had been changed.

This was because of the way extra nodes on rigid parts were being drawn and this has been recoded to make it much faster.

The Sketch operation in this context will now look at the 'Stop' button periodically, so if a sketch is still taking a long time the user can terminate the operation by using STOP.

Connections

- **Case 26152**

Filtering the object menu for connection by connection title could result in SIGSEV. Now fixed.

- **Case 25319**

The description for constants SPOTWELD_SOLID in the JavaScript API manual has been corrected to signify solids/spotweld layer.

- **Case 25021**

When creating tied contacts for connection entities, a penalty based contact is often created in parallel to a constraint based one to ensure any nodes that do not tie due to a constraint clash are tied. PRIMER v11 created *CONTACT_SPOTWELD_OFFSET when creating a penalty contact in parallel to a *CONTACT_SPOTWELD contact, but _OFFSET cannot be applied to a SPOTWELD type contact. PRIMER will now create a *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_OFFSET contact instead.

- **Case 24887**

When auto-creating connection information on top of existing spotwelds, PRIMER did not consider the panels any spotwelds were meshed into when populating layer information for the connection entity. This could lead to a connection entity with unexpected layer information, and therefore labelled as BAD. This has been corrected.

- **Case 24595**

Connections data can now be written out to IGES format file from the command line.

- **Case 24454**

The 'Ignore parts' option on the connection read panel did not ignore parts as intended. This has been corrected.

Constraints

- **Case 25497**

Creation of NRBs using screen area pick with a large number of nodes was found to be very slow. Now fixed.

Contacts

- **Case 25910**

Illegal floating value error messages no longer appear if Check > Options input boxes contain the valid or strings.

- **Case 25737**

PRIMER gave an error by mistake if creating/checking a *CONTACT_2D_AUTOMATIC_SINGLE_SURFACE card without specifying the SIDM field (not required for the single surface contact type). This has been corrected.

- **Case 25706**

Null Box Id field when set type (MSTYP/SSTYP) doesn't support Box definition.

- **Case 24664**

The MPP method switch for penetration check under check > contact > options did not work resulting in the default MPP method being applied always with possible incorrect report for SMP contacts. Now fixed.

- **Case 23718**

PRIMER now considers *ELEMENT_SHELL offsets during contact penetration checking.

Note that LS-DYNA, and hence also PRIMER, only consider offsets during the contact penetration calculation if:

Field CNTCO is set on *CONTROL_SHELL.

The contact surface is not an AUTOMATIC type.

In addition nodes on the slave sides of NODES_TO_SURFACE contacts do not 'inherit' the offset of their parent shell.

(See also case 15950)

- **Case 20622**

Error checks that determine whether nodes lie on or behind rigidwalls now consider the effect of SHLTRW in *CONTROL_CONTACT.

Contour

- **Case 25113**

Made the following changes:

When ESORT=1 on *CONTROL_SOLID, and you do a contour plot of ELFORM,

- If the ELFORM on *SECTION_SOLID is 19 and the element is a wedge (PENTA), displayed as 21

- If the ELFORM on *SECTION_SOLID is 20 and the element is a wedge (PENTA), displayed as 22

- A TETRA with ELFORM=19,20,21 or 22 is omitted and throws an warning.

- **Case 23999**

Bands on the contour ramp for all relevant CT-SI plots now respond to cursor-driven actions.

Left clicking on a particular band restricts the contour to entities corresponding to the current band. It is possible to quickly interrogate contours corresponding to different bands by left-clicking on various bands successively.

Middle clicks reset the contour.

Right clicks reveal additional contour refinement options.

- **Case 15071**

Timestep for 4 noded tet elements with formulations 16, 17 was incorrectly calculated. Now fixed.

Control

- **Case 25308**

-Sorted out non-static control cards order to be alphabetical.

-The type code for CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS was CMDAP, change it to CMDP.

- **Case 23985**

Changing the field TYPE (part or part set) to a value which is out of range was crashing the PRIMER, now TYPE is set to default(PART type) if it is out of range.

Curve menu

- **Case 21755**

When importing 'xy' data in the loadcurve editor the file selector will now show .txt files and .csv files (previously only .txt files were shown)

Cut section

- **Case 25879**

The sequence:

1. Read model
2. Define and turn on a cut section
3. Turn off cut section
4. Delete model
5. Read new model
6. Turn on cut section

Could cause the new model not to be displayed in cut section form until some explicit graphics command was issued.

This was because the management of the cached graphics for cut sections wrongly believed that the new model was up to date, whereas it needed an explicit redraw. Now fixed by better cached graphics management.

- **Case 25794**

The cut section property calculation will derive Area, centroid, Ixx, Iyy and Ixy properties for cut sections, and these are calculated correctly. However when calculating properties for a doubly symmetric equal section such as a cube or a circle, where Ixx == Iyy, problems could arise when reporting the angle between the principal axes Iuu and Ivv and the element axes Ixx and Iyy.

Clearly in a double symmetrical equal section where Ixx == Iyy then the off-diagonal term Ixy should be zero, and the principal terms Iuu and Ivv should be equal to Ixx and Iyy. This also means that the angle theta between the principal terms Iuu/vv and the element axis terms Ixx/yy is indeterminate and should be treated as zero.

In real life such sections, when made up from finite elements and nodal coordinates, tend to have small dimensional errors which mean that there is a small residual Ixy term, and also that Ixx and Iyy may be slightly different. PRIMER knows this and has a tolerance check to detect this and to clamp the angle theta to zero, but this was set too tightly and did not pick up the ill-conditioning case reliably enough.

The tolerance check has been improved so that not only does it look at the radius of the Mohr's circle, but also now also considers the relative magnitudes of Ixx/yy and Ixy.

A separate problem with the optional calculation of the torsional constant J for a thin-walled closed section made of shell elements was also detected and corrected.

- **Case 21706**

Certain beam types, notably elform = 2 and elform = 12, have their engineering properties supplied as explicit Area, Iss, Itt and Irr values, and do not have shape information.

When 'true' beam section display is used PRIMER attempts to generate a shape that gives a reasonable indication of the beam's size and orientation. This cross-section shape is also used when calculating section properties in the cut-section panel.

Historically this has been calculated as a rectangular hollow section, and this usually works, but can give poor results in some situations:

+ The properties are non-physical. For example the quoted area is not compatible with the quoted Ixx and Iyy values even when a solid rectangular section is used. (Such properties can be correct if they represent a composite section, but there is no way of knowing this.)

+ The properties can be represented as a thin-walled rectangular hollow section, but its aspect ratio becomes ridiculous because they actually refer to an I beam.

+ The torsion constant, J, is not considered when synthesising a section. Therefore if torsion constants are to be computed in the cut-section properties panel they are likely to be wrong.

Various improvements have been made:

- The synthesised cross-section will now be an I beam if the engineering properties make this shape seem more plausible than a rectangular hollow section. The switch-over point between these two section types is when the RHS height : depth ratio exceeds 2:1 or 1:2.

- Better error trapping for 'non physical' properties has been added, with reversion to the best achievable solid rectangle being used rather than a 'hollow' section in which the walls overlap.

- In the cut-section property calculation the supplied engineering properties (Area, Iss, Itt, Ist, J) are used explicitly, rather than attempting to calculate properties from the synthesised section shape.

Database

- **Case 26122**

Sometimes the Create button could be greyed out when creating *DATABASE_CROSS_SECTION cards using the 'feature line' mode. This has been corrected.

- **Case 25292**

Picking nodes for NID when there is no Title in DATABASE_HISTORY_NODE_LOCAL_ID no longer causes crashes.

- **Case 25054**

Set Defaults button for *DATABASE_CPM_SENSOR now works.

- **Case 24970**

Model check now reports correct number of Database categories when DATABASE_EXTENT cards are present.

- **Case 24935**

Editing the DCOMP field (overtyping rather than using the pop up) in *DATABASE_EXTENT_BINARY now works correctly.

- **Case 23478**

Listing for DATABASE cards has been tidied up.

Deleting

- **Case 25372**

A model containing *DEFINE_CONNECTION_PROPERTIES, in which reference is made to many materials, but where the card itself is referred to by other materials, failed to 'clean up'.

This was really a more generic problem that MATerial is an 'owner' category when it comes to selection (blank a material and you expect to see its parts and elements disappear), but a 'child' category when it comes to deletion checking. Adding some extra logic to make this distinction in the right places fixed the problem.

- **Case 23674**

Modified the part deletion logic in relation to DEF_TO_RIGID cards.

- **Case 17653**

The 'Remove, Cleanup' operation has special settings for empty PART and SET cards, the default being that if these are found they are marked for removal since they are redundant.

However it is possible to change the default behaviour to 'no action' which should leave both parts/sets and references to these unchanged in this situation. This works correctly where references to parts/sets exist, but if no such references exist then the parts/sets are still marked for deletion.

This is inconsistent behaviour and has been fixed: where *SET or *PART cards exist, but these have no contents and are also not referenced then if 'no action' is chosen then they will no longer be marked for cleanup.

Dialogue command

- **Case 25483**

When using the command line to create LS-DYNA seatsquash models you could not turn the global damping option back on after you had already turned it off. This has been corrected.

Dummies

- **Case 25952**

Historically PRIMER has marked 'stop angle' values on *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED that lie outside the range +/-180 degrees as an error.

This is incorrect - or at least overly conservative. Since LS-DYNA keeps track of the number of rotations a joint undergoes during an analysis there is no need to restrict stop angles to any particular value, and in fact angles greater than 360 degrees might make sense.

However during initialisation of Joint Stiffness definitions at time zero the *initial* angle can only be evaluated in the range +/-180 degrees, so it makes sense to clamp the initial orientation of these definitions to that range. Therefore the following changes and additions have been made:

- The error message for a Joint stiffness stop angle outside the range +/-180 degrees has been removed from the joint stiffness check function.
- Input of explicit stop angles in the joint stiffness editing panel and keyword editor will now accept values outside the +/-180 deg range.
- The dummy positioner will now allow stop angles outside the +/-180 deg range to be input, but issues a warning that during positioning PRIMER will clamp movement to +/-180 deg to prevent 'wrong side of zero' errors during initialisation in LS-DYNA.
- Dummy checks now include a test for a given initial orientation being outside the initial stop angle range, and also for stop angles outside the +/-180 deg range when articulation about more than one axis is permitted.

- **Case 25837**

With Dyna Dummy Positioning and Dyna Seatsquash, if the dummy contains type 6 beams, it is possible that the resulting LS-DYNA analysis would not run in LS-DYNA, due to the type 6 beams not being deleted during the rigidification process when they could be. This has been corrected.

- **Case 23726**

When tying in values for the 'neck' or 'hip' node during 'dyna' dummy positioning it was possible to type in a node that did not exist in the model. This has been corrected.

Edit

- **Case 23736**

The *ELEMENT_BEAM edit panel will now remember settings such as keyword options and values for fields vx across multiple 'Create' instances as long as the panel isn't dismissed.

- **Case 23576**

In *BOUNDARY/*CONSTRAINED keyword if you go to any keyword panel(eg. BOUND->ACOUSTIC_COUPLING->LIST->LIST_ALL) and select LIST_ALL, it lists all the keywords which use sub-types. Now it is fixed.

Elements

- **Case 25831**

PRIMER was not calculating the correct mass for *ELEMENT_DISCRETE_SPHERE_VOLUME elements. Now fixed.

Errortrapping

- **Case 25135**

When deleting a model that contained *COMMENT keywords PRIMER did not return the memory for the comment correctly and errors occurred. Now fixed.

- **Case 24867**

A spurious error message was given if a node was edited and renumbered from the database history node panel. Now fixed.

- **Case 23774**

Crash in primer while editing CONTROL_FORMING_ONESTEP card. Now fixed.

Geometry

- **Case 25936**

PRIMER failed to read a specific IGES file due to a character string problem. Now fixed.

- **Case 24892**

Geometry tessellation has been improved. Fewer errors should now be given for surfaces.

- **Case 24534**

If geometry was oriented then cut sections using omit for both sides of the cut would not draw anything. Now fixed.

Graphics

- **Case 25005**

Graphics could not get refreshed after using the node keyword editor. Now fixed.

- **Case 24616**

Previous versions of PRIMER would not reset local element axes visibility and associated options when the shortcut key 'k' was pressed. This has now been fixed.

- **Case 22220**

When setting the 'display factor' in any Oasys Ltd software it is possible to choose a value that makes the user interface of the software unreadable.

To prevent this happening there is now a confirmation dialogue box - at the unchanged scale value - that requires you to confirm that the revised setting is to be kept. It also offers the option of reverting to the original factor.

Image

- **Case 24330**

From Windows Vista onwards the 'Aero desktop' changes the way that pixel data in images is created and stored, and one side-effect of this is that the existing 'Copy to clipboard' function available from the top left [-] popup menu on all sub-windows fails to read 3d graphics generated by OpenGL. As a result using this on a graphics window tends to leave a blank space where the picture of the model should be, although the window surrounds will be drawn correctly.

Capturing ordinary GUI windows works fine, and the various options to generate files under 'Image' also work correctly.

This has been fixed in V12 by performing the extra work necessary to capture the 3D graphics when copying window contents to the clipboard.

Include

- **Case 25943**

PRIMER could give an incorrect error message about entities being outside specified include ranges when running a model check on a model containing *ELEMENT_MASS_PART entities and containing PRIMER label ranges. This has been corrected.

- **Case 21845**

Include declash now allows the user to merge duplicate nodes (and keep the duplicate).

- **Case 21449**

Historically PRIMER has permitted the category INCLUDE FILE to be selected for deletion, and doing so will select all its contents correctly, however the include file itself will not be deleted. The same is true if MODEL is selected for deletion. An annoying consequence of this is that *DEFINE_TRANSFORMATION cards referenced from *INCLUDE_TRANSFORM can therefore not be deleted, and in addition any *PARAMETERs used on these cards will also be locked against deletion.

This is because Include files inside PRIMER are a special category that did not respond well to the normal generic processing used for deletion, meaning that they could not be deleted this way.

To fix this:

+ a new option 'Empty include file action' has been added to the Delete and Cleanup removed panels.

+ this defaults to 'No action', meaning leave an empty include file, which is no change to existing behaviour.

+ however the user can choose to 'Delete empty file', in which case the include file itself will be deleted if empty.

Include transform

- **Case 26124**

Primer could crash when read a model that contained *INCLUDE_TRANSFORM's which referenced *DEFINE_TRANSFORM's which had POS6P or POINT based ROTATE type transforms and the include file referenced in the transform has nested includes. This has been corrected.

- **Case 24565**

*INCLUDE_PATH definitions in INCLUDE_TRANSFORM include files did not work. Now fixed.

Installation

- **Case 23994**

PRIMER opens temporary files for various purposes, and on Windows platforms these will be called pr_tmp_nnnn where nnnn is a number in the range 1 to 9999, and will reside in the user's home directory, or in the OA_TEMP directory if that is defined.

If PRIMER crashes, or is killed in an abnormal fashion (for example by killing the controlling terminal window) then these files do not get cleaned up, resulting in a build-up of files in that area. If the limit of 9999 files is reached PRIMER will generate error messages on start up.

The workaround is simple enough: just delete the files, but this should not be necessary. Therefore the temporary filename allocator now has some logic that inspects the age of existing files pr_tmp_nnnn, and if they are older than the stipulated expiry date - by default 31 days - then the file is deleted and the name re-used.

The expiry date can be modified using the preference

oasys*temp_file_expiry: n

Where n is a number of days in the range 1 to 10,000. If n is set to zero then testing for file expiry is switched off.

Keyword

- **Case 25119**

*ELEMENT_SEATBELT_PRETENSIONER problem with PTLCID - fixed it by changing ptlcid=NULL when sbprty=2,3

- **Case 24554**

*LOAD_BODY_VECTOR is now supported by PRIMER.

- **Case 24542**

Checks added to *DATABASE_FSI, *DATABASE_FSI_SENSOR and *DATABASE_CPM_SENSOR.

- **Case 24477**

Made change so that FTYPE for *INTERFACE_SPRINGBACK can now be 0,1,2,10,11,12.

- **Case 24283**

Data for EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE that is consistent with LSDYNA version 950 and later can now be input.

- **Case 24144**

When a part uses section point source, it is correct to define *EOS card. PRIMER incorrectly removed the reference to this card when the part edit panel was invoked. Also we incorrectly gave warning when such a part had no elements.

- **Case 23884**

Changing the field TYPE (part or part set) to a value which is out of range was crashing the PRIMER, now TYPE is set to default(PART type) if it is out of range.

- **Case 23805**

Sorted errors related to Keyword editors of *MESH_VOLUME, ICFD_BOUNDARY.

- **Case 23661**

Fixed: *CESE_MAT_GAS not read by PRIMER.

- **Case 23660**

Fixed: *CESE_PART not read by PRIMER

- **Case 23491**

Fixed it with case 22937, at least one solid element should be selected to create hex spotweld.

- **Case 23353**

Fixed: labeling issue with *ELEMENT_DIRECT_MATRIX_INPUT cards.

- **Case 23095**

PRIMER ignored large memory (e.g. 900000000) as memory allocation, and shows garbage value in *CONTROL card. Now fixed

- **Case 22296**

Fixed: PR12 fails to read *CONTROL_DYN_RELAX card.

Kwd editor

- **Case 26142**

If the keyword editor is asked top display data for cards with many rows, in this case *INITIAL_STRESS_SHELL with

- 4 integration points on plan

- 5 integration points through the thickness

- 97 extra variables per integration point

So a little under 200 values per element, then it did not set aside enough buttons to display more than 2 cards worth of data, even when using 'all on one row' format.

This limit has been raised so that at buttons for at least 40 rows are allocated

Labels

- **Case 24798**

Label ranges were not obeyed during multiple split if a shell was already outside the valid label range. Now fixed.

- **Case 24584**

When changing the include file on an edit panel while creating an entity the label of the entity could be set to zero. This has been corrected.

- **Case 24033**

In the ENTITY panel, where what is drawn is controlled, it is possible to choose what a labelled item is 'Labelled with'. This can be more than item's label, extending to its model, material id, etc.

For entity type NODE these 'labelled with' switches were not working properly, meaning that the graphics might not be updated correctly to show the current setting. This has now been fixed.

Local axes

- **Case 25006**

Local material angles for shells defined with the _MCID keyword option could be shown incorrectly in certain situations. This has been fixed.

Macro

- **Case 25146**

If a macro was run from a JavaScript and PRIMER was started with either the -batch or -exit command line arguments PRIMER would exit when the macro finished instead of when the script (that called the macro) finished. Now fixed.

- **Case 23986**

When playing a macro with Pause commands 'Button box not yet defined' error messages were occasionally shown. This has been fixed.

Mass

- **Case 25392**

The count and mass of included part/nrb _inertia displayed on the assign mass panel was incorrect if sub-group massing was in use, though the calculation was itself correct. Now fixed.

- **Case 25089**

When using the 'mass prop' feature, it was possible to get the wrong answer when selecting *ELEMENT_MASS elements. This has been corrected.

- **Case 24551**

Selection of an include file in the mass property calculator resulted in inclusion of any lumped mass on the nodes of the include, irrespective of whether or not the element from which the mass derived is in the selected include. This has been fixed so that, unless the option is set to include attached mass on elements, mass derived from elements outside the selected include will be excluded.

Materials

- **Case 25347**

PRIMER could crash if importing a material to a material edit panel, and the 'Filter' option is used. The crash would only occur if clicking on the 'Filter' button before typing in the filter string. Now fixed.

- **Case 24487**

Fixed: The field SBUCK for MAT_203 has a default and therefore zero value is now allowed.

- **Case 22284**

Fixed the units of fields & of MAT 196.

Mechanism

- **Case 24871**

Mechanisms in PRIMER provide several ways of 'driving' motion, and one of these is to define a new angle for a HINGE or LINE joint. The mechanism positioner will rotate the attached assemblies to achieve the desired angle.

In a few cases the direction of rotation went the 'wrong' way, in the sense that if clockwise rotation was requested what was actually produced was anti-clockwise motion.

This was due to an error in calculating what constitutes positive motion of a body about a vector: the existing algorithm could fail in cases where the orientation of the bodies was not well-conditioned.

The correct calculation method has been substituted and rotation should now take place reliably in the requested direction.

- **Case 24857**

(This case is related to case 24871).

There was a report of mechanism movement 'hanging', or refusing to achieve the desired angle, when using the feature that allows rotation motion of a joint to be 'driven' to a new angle. We don't have the model to test.

A very similar bug, reported in case 24871, in which driven rotation went the 'wrong way' has been corrected. The cause of this was a failure to calculate the correct rotation angle and direction for 'driven' motion, and it could produce symptoms identical to those described in this case.

Because we don't have an example model for this particular problem we cannot say with certainty that it has been fixed, but it seems very likely that it will have been, so this case is being closed.

Memory

- **Case 25991**

Testing revealed that the internal allocation of memory used a strategy that was 'too small' for very large models, meaning that when these were deleted the process of returning memory to the operating system could cause PRIMER to 'hang', or at least to take a very long time.

This problem would only show up for models with more than around 20m nodes and elements, which is beyond current production practice, so it is unlikely that it will have been encountered during normal use.

The memory allocator has been modified so that it adjusts automatically to this very large models, meaning that model deletion should now run at a reasonable speed. No user intervention is required.

- **Case 25803**

PRIMER did not return memory for *BOUNDARY_PRESCRIBED_FINAL_GEOMETRY cards correctly which meant that if a model containing lots of these cards was read and deleted many times PRIMER could use excessive memory. Now fixed.

Menus

- **Case 25349**

Running PRIMER with the command line '-batch' option (note, not the same as -d=batch) does the following:

- Starts the graphical user interface and normal graphics

- Suppresses the display of the master PRIMER window

In other words it is running normally, but you don't see the window.

For historical reasons this also turns off the 'menu auto confirm' logic which gives the default response to any 'stop and confirm' messages.

This means that anything which stops and asks the user for a response will lock up PRIMER since the window with the message, and the button you would click on to continue, are not visible.

Therefore a new command-line option '-auto_confirm' has been added which works as follows:

- If, and only if, used in conjunction with '-batch' it turns on the automatic confirmation of the default response at these 'stop and confirm' panels, meaning that execution will continue.

- In all other contexts, ie if '-batch' is not also used, it is totally ignored.

Merge

- **Case 25705**

When merging nodes if both nodes had *DATABASE_HISTORY cards then PRIMER would leave the duplicate after merging. The *DATABASE_HISTORY card on the slave node is now deleted.

- **Case 24929**

The option to collapse quads to trias did not work in node merge. This has been fixed.

- **Case 24338**

In Remove, Merge nodes there was poor ergonomic behaviour in the following circumstances:

Database contains two models, only one of which is active.

Merge nodes is in use in the active model.

The user makes the other model active in Model, List

The user found that he could not select items from the newly active model for merging, since the still open selection menu applied only to the original model. Worse, clicking on MERGE again did not reset this.

A trap had been added to ignore 'click on Merge if already merging', since to do so would restart the process losing all items currently selected for merge, and it was assumed that a second such click would be accidental.

This logic has been changed to a warning message explaining what restarting the merge process entails, and giving the option to cancel or continue the process.

In addition the status messages given in the feedback buttons have been amended to show which model is currently being processed, since this was not originally shown.

(All the above does not change the underlying functionality of Merge)

Mesh

- **Case 25718**

If a quad was split into 4 quads the local axes for the 4 shells created were not consistent. Now fixed.

- **Case 24673**

A crash has been fixed in cobweb mesh.

- **Case 24600**

For 'element detach', when the 'auto' mode was selected, detach did not work if the auto number was set to 1. This has been corrected. Auto mode also did not work for solid elements. This has also been corrected.

- **Case 22617**

In extrude shell panel, drop-down says 'no of shells', now updated to 'no of solids'.

Metal forming

- **Case 23758**

Inconsistent behavior with 'Part Selection' on Tool->FORMING menu, part selections were retained if Model is changed using text box but reset if Model is changed via Model selection panel. Now it's fixed.

Model build

- **Case 26119**

IHI style build-from-csv, when used in conjunction with the AUTO vertical option, was writing INITIAL_VELOCITY_GENERATION in a cumulative fashion. In other words, the first master model would get one card, the second would get that plus another card and so on. This has now been fixed.

- **Case 24543**

Build from csv in modes where the input is only XY data did not calculate the corresponding Z value very accurately on models where the mesh is coarse. The calculation has been improved to use the proper parametric formulation and will now give an accurately interpolated value.

- **Case 24304**

A redundant 'Options' button has been removed from the 'Save Database' panel.

Model modified

- **Case 25040**

The part compare option 'Select parts from modified model' did not work correctly. Instead of comparing just selected parts it compared all parts. This has been corrected.

- **Case 24364**

For some open-ended keywords, Model Modified failed to identify that fields have been changed. Now fixed.

- **Case 22210**

Model modified fails to report changes to expressions in *DEFINE_CURVE_FUNCTION. This has been corrected.

Nastran

- **Case 25658**

Fixed faulty logic of appending .dat at the end of include files.

- **Case 23760**

PRIMER could give warnings if multiple Nastran files were read. Now fixed.

- **Case 23689**

PRIMER was crashing when reading in a Nastran file which had something like 'TITLE =' line. Now fixed.

Nodal rb

- **Case 23998**

When creating NRBs and other constraints using 'Pick N closest nodes' or 'Pick with tolerance' PRIMER will now select the closest nodes in 3d (model) space. The previous use of 2d (screen) space could result in large constraints accidentally being created through the depth of the model.

Additionally the test for node visiblity in Hi/Shaded mode has been made rigorous.

- **Case 23579**

The PRIMER 'Cleanup' tool will no longer offer to remove 2-noded CONSTRAINED_NODAL_RIGID_BODY, _SPOTWELD, and _RIVET definitions that connect 'centre' nodes of other NRB spiders.

Node

- **Case 25776**

The node edit panel includes an option to create nodes at the centre of a hole. The logic used for this function was producing incorrect results when nodes were clustered around one side of the hole. This has now been fixed.

Orient

- **Case 25359**

Interpolation option is should not be available for Orient->translate/rotate/... with 'Copy' on - geryed out the buttons accordingly.

- **Case 25358**

Orient=>Translate (with interpolation on, and 'part only' selected) when applied to nodes, the interpolation effect is not seen on the parts attached to nodes. Now fixed.

- **Case 25093**

Fixed: Undo is greyed out in confirm Orient panels after doing project to line and trans-rot functions.

- **Case 24552**

Rotating an *INITIAL_STRAIN_SHELL card using orient could corrupt data and cause PRIMER to crash. Now fixed.

- **Case 24546**

Orient with interpolation of nodes applied to more than one model in the same orient session could result in operations on one model affecting the nodes of another. Now fixed.

- **Case 24354**

Added display of individual and cumulative movement of distance translated/rotated/scaled in orient panel.

- **Case 24353**

Sketching of selected nodes disappeared after orient, even though they were still selected. Now fixed.

- **Case 24340**

The 'Orient' menu has a 'Node Drag' button. Clicking on this button will no longer close the 'Orient' menu.

- **Case 24312**

Orienting a model with the copy option turned on did not copy contacts if the 'elems in the same part' option was used. Now fixed.

- **Case 22408**

When orient copy is applied to a part in an include with the option to make a new part, the section and material cards are automatically copied as well. The new part will be in the same include as the original but the section and material were ending up in the master file. Now they will be in same include as part.

- **Case 22339**

Orient copy with make new part option has been fixed to work for latent parts. Previously a proper installed part was required.

Parameter

- **Case 25128**

When the additional mass on a part, defined by the *ELEMENT_MASS_PART card, was defined by a parameter the calculated mass of the part was not updated if the value of the parameter was changed.

Caused by the routines that update the parameter not 'telling' the model that some cached data might have changed, resulting in the old (cached) mass value continuing to be used.

Fixed by making sure that this flag is updated whenever a parameter that is referenced has its value changed.

- **Case 25001**

When a model contains many parameters the popup list of eligible parameter names that appears when you type '&...' into a text box can get very large. Prior to V12 there was no way of managing this popup if it was larger than the enclosing window.

This popup (and others in similar contexts) will now have scroll-bars if they are larger than their parent window.

In addition it is now possible to use wild-card syntax when entering parameters into a text box. '?' replaces a a single character and '*' replaces multiple characters, and by using this syntax you can control the list of candidate parameter names displayed in the popup listing.

- **Case 24525**

The operation INCLUDE, new child could go wrong if:

+ The destination model contained parameters

+ The new child include file referenced those parameters

During the import operation the new include file is temporarily read into a scratch model prior to checking and import into the destination model, and if this scratch model does not 'know about' parameters in the destination model then errors can arise.

This has been fixed by copying all all parameters in the destination model into the scratch model prior to reading the new include file so that references to them can be resolved correctly. Special logic stops them triggering label clash checks, and also prevents their being copied back into the destination model.

- **Case 23356**

IN a PARAMETER_EXPRESSION PRIMER will accept:

either A^B

or A**B

syntax to mean 'A to the power of B', but LS-DYNA will only accept the Fortran syntax of A**B.

This is now detected as an error while checking parameters, and there is an autofix to replace '^' with '**'.

(PRIMER will continue to accept either syntax so that it remains able to read and evaluate these definitions, as rejecting them during input could cause problems.)

- **Case 16682**

When PRIMER encounters a parameterised field before the true value of the parameter is known it adopts the following policy to cope with this:

+ A value of zero is inserted as a temporary parameter value, unless the usage is in a label field.

+ In a label field a special internal 'unknown label' value is used instead.

+ When keyword input is complete and true parameter values are now known PRIMER goes back and rebuilds the internal data of the cards in question using the correct values.

This process works well so long as the use of zero does not invalidate the card format in any way, ie is illegal, or the field in question does not dictate the format of the rest of the card.

For example some *AIRBAG and *MAT cards have a variable number of rows that is dictated by earlier data fields, *PART_COMPOSITE needs to know how many layers are to follow, and so on. If the key data fields in these situations are temporarily assigned zero then the rest of the card will be read wrongly, and no amount of subnsequent rebuilding from internal data will work.

From V12 onwards there is an option under READ, OPTIONS to 'pre-scan' the input deck for parameters. This works as follows:

+ The input files are scanned and *PARAMETER cards are read, but all other cards (other than *INCLUDE to access include files) are ignored.

+ This means that all values of parameters are then 'known'.

+ Then the deck is reread properly, ignoring *PARAMETER cards since these are now known, and correct parameter values can be substituted immediately in fields where they are used.

This solves the 'used before defined' problem, but it comes at a cost: the pre-scan for parameters is faster than a normal read, but still takes a significant amount of time, which is why it is not the default behaviour. However it is the recommended solution for decks that will not otherwise read correctly.

Note that PRIMER V12 will always endeavour to write out parameters before they are used, so it is possible that reading in and writing out such a deck may solve the problem for future read operations by moving *PARAMETER keywords to earlier in the deck.

This cannot always work, for example where *PARAMETER cards are placed in an include file, and this include file is read after other include files which reference the parameters, the problem will remain since PRIMER does not automatically resequence include files.

Part replace

- **Case 24981**

The current include could be changed randomly after a part replace, and when closing the main part panel. This has been corrected.

- **Case 24036**

Operations such as part replace which create primer connections from existing welds, have been modified so the creation is only performed in the local area rather than across the whole model.

- **Case 23690**

When doing an assembly replace with a target model containing latent nodes which exist at the same label in the source model, it was found that these nodes are incorrectly merged, causing unwanted connection to an nodal rigid body. The source model is now renumbered to avoid clashes wrt latent items as well as installed ones.

- **Case 23418**

When part replace is applied to a single solid part, the process deleted any coating shells on the target part and imported any on the source part. If the source part had no coating shells this could result in unwanted shell deletion, so now no action is taken if the source part has no coating shells.

Part tree

- **Case 26086**

Changing the colour of a Primer assembly (for example through the part tree) did not work. This has been corrected.

- **Case 25911**

When the mouse is hovered over an item in the Part Tree that is capable of being sketched, then PRIMER will highlight it in the graphics window. (Logic similar to predictive picking.)

This works correctly, however when the mouse is moved out of the Part Tree window the highlighted item is not 'un-sketched', which can be undesirable, particularly if - as in the case of an include file -sketching has drawn many things.

This has been corrected: moving the mouse out of the Part Tree window will 'un-sketch' anything that has been highlighted in the graphics window.

- **Case 25375**

Cutting and pasting DAMPING_GLOBAL and DAMPING_MODAL in Part Tree Contents mode now works.

- **Case 25301**

Cut and paste LOAD_BODY in part tree (contents mode) now works.

Pedestrian

- **Case 26043**

The pedestrian markup script will now calculate offset lines by default.

- **Case 25967**

The pedestrian markup script could fail if the vehicle parts selected meant the upper and lower bumper lines could not be created. This has been fixed and a warning message is now written if this is the case.

- **Case 25914**

The pedestrian markup script would fail to build models if CSV file specified by the user contained spaces in the pathname. This has been fixed.

- **Case 24972**

The pedestrian markup script would crash if the output csv file used a pathname that contained directories that didn't exist. It now creates the directories so the file can be written.

- **Case 24672**

In the pedestrian markup script you can set the requested beam length for creating the lines. This was really the 'grid size' between sections through the vehicle, but if a beam was created at a large angle to the X or Y axes then the beam could get long, resulting in a loss of accuracy.

This has been improved in v12 so that beams are created with the requested beam length.

- **Case 24499**

In the pedestrian markup script, if you tried to write the impact points to a CSV file that was open in Excel the script would crash. This has been fixed.

- **Case 24496**

If the pedestrian markup script can't reach a WAD length a warning message is displayed giving possible reasons why it failed. An additional reason has been added to explain that the X axis should point backwards.

- **Case 23992**

The pedestrian markup script for the EuroNCAP grid protocol was not moving the location of grid points in the bonnet-windscreen gap as described in section 3.9.6. It now does.

Penetrations

- **Case 24876**

PRIMER ignored STF on the *PART_CONTACT card when doing penetration checking. This has been fixed.

Preferences

- **Case 23654**

In PRIMER (and other software) it is possible to override the default directory in which checkpoint files are written by using the

primer*checkpoint_dir: (also oasys*checkpoint_dir: and so on)

preference. This can also be done by setting the environment variable CHECKPOINT_DIR.

The only problem was that it didn't work! The logic to set that environment variable internally from the preference was not working correctly.

This is now fixed. However users should note that if an alternative directory is defined then it must exist, it will not be created. If it does not exist, or is read-only, then checkpoint files will continue to be written in the default location.

Properties

- **Case 25571**

PRIMER could crash on Linux when clicking on the 'Import from file' button on the properties store/save panel. This has been corrected.

Quick pick

- **Case 25723**

When using quick pick to type in entities you wish to ONLY sometimes it did not work if the entities you types in were currently blanked. This has been corrected.

Read

- **Case 25345**

Models using *INCLUDE_PATH_RELATIVE allow a relative path for include files to be added, but this pathname was not being dealt with correctly either during normal keyin or during the various 'scan' operations.

Now fixed.

- **Case 24643**

PRIMER could hang when reading an IGES file if some of the surfaces contained corrupt data. This is now fixed. Additionally offset surfaces (IGES type 140) are now supported.

- **Case 24521**

Model scan would fail if the include files contained duplicate control cards. Now fixed.

Renumber

- **Case 25351**

When selection the 'master include' to renumber using 'Renumber selection' or 'Clipboard renumber' the resulting panel which should list all entities within the master file was empty. This has been corrected.

- **Case 25078**

PRIMER has a check for parts with the same ID and NRB's, and also has an autofix to relabel the NRB's to remove the clash. The autofix gets a new label using the current include layer, whereas it should use the include layer that the NRB lives in. This has been corrected.

- **Case 25057**

Relabeling in the 'Renumber selected' panel could lead to duplicate labels in entities if changing from 'Start from' to 'Offset' after specifying the start/offset value. This has been corrected.

- **Case 24520**

A new preference and a program option have been added that can prevent renumbering of node sets used by 2D sliprings into values of eight digits or more. Such node sets are legal in more recent LS-DYNA versions (971R7.1+). Therefore, the new option only comes into effect for versions R7.0 and older.

- **Case 24142**

Keyword *ALE_MULTI-MATERIAL_LIST is unusual in LS-DYNA in that it does not have an explicit label, but each row of entries within it becomes implicit label i, where i is the order of appearance. This implicit label is then used to reference that row on other *ALE cards, *DATABASE_ALE and *SET_MULTI.

As a consequence PRIMER treats *ALE_MULTI-MATERIAL_LIST definitions as being 'labelled' because this is true. However unlike other items referred to by label these definitions require special treatment which 'protects' them from renumbering, since a model containing N such definitions must always label them 1 to N.

This special treatment was missing, with the result that a user could relabel them in some circumstances (eg Model, Renumber) and this could cause all sorts of problems, including

'hanging' PRIMER if the increment was very large. This was because, in order to preserve the 'label is order of appearance' rule, PRIMER would create the intervening definitions as latent entries.

This has been fixed by recategorising these cards internally as 'not explicitly labelled', despite the fact that they do effectively have labels that are used to refer to them.

This will protect them against conventional renumbering, however it will not protect them against accidentally typing in a large label when referring to them on other *ALE cards, *DATABASE_ALE or *SET_MULTI. Users should exercise care in this respect.

- **Case 23954**

Logic has been added so that under normal operation PRIMER will avoid giving *PART cards and *CONSTRAINED_NODAL_RIGID_BODY cards clashing labels.

- **Case 23940**

When creating section or material which uses a character string for its label in a model with include label ranges defined, the checking function incorrectly reported the items as having out of range labels. PRIMER now excludes these from the label range check.

- **Case 23069**

The include numbering table no longer warns about out-of-range database history cards and their children when the option to lock database history cards against renumbering is invoked.

Rigidify

- **Case 24303**

Rigidify could leave errors in the model if the parts to rigidify used 'chained' nodal rigid bodies or if they contained beams which had 3rd nodes in a nodal rigid body. Both these issues have been fixed.

Scripting

- **Case 26080**

Various panes or tabs in the FMH script now automatically resize to optimum dimensions

- **Case 25721**

Part.ClosestNode() would crash if used with a beam part if some of the beams had less than 3 nodes. Now fixed.

- **Case 25463**

The LCO option for *ELEMENT_DISCRETE was not supported in JavaScript. It has now been added.

- **Case 25394**

In very rare cases the debugger crashed when pressing Next or Step. A preference has been added to turn off JIT (Just In Time) compiling if this happens for other scripts.

- **Case 25365**

Character fields (e.g. the ALIAS/RE field on *MAT_RIGID) could not be set from JavaScript. Now fixed.

- **Case 25289**

Set.RemoveGeneralData() and Set.SetGeneralData() did not update the set cache for graphics properly so the wrong things could be drawn. Now fixed.

- **Case 25055**

Shell free edges were not available in JavaScript if PRIMER was run in batch mode. This is now fixed.

- **Case 24874**

The JavaScript function UnblankFlagged() did not work as expected for some entity types, such as Materials. This has been corrected.

- **Case 24684**

PRIMER could crash if the heading property for a joint was set without the id property being true. Now fixed.

- **Case 24659**

PRIMER had a global Error() method to write an error message to the dialogue box. Error is a reserved word in JavaScript and this is causing problems when used with some public JavaScript libraries. In version 12 the method has been renamed to ErrorMessage().

- **Case 24658**

The debugger could crash if a script contained large numbers of nested functions (functions inside functions). Now fixed.

- **Case 24639**

The JavaScript debugger could crash if a script had extremely long lines. Now fixed.

- **Case 24564**

When reading an LSTC dummy using the LSTC dummy conversion script, part of the dummy title could be lost. This has been corrected.

- **Case 24549**

Button clicks in JavaScript windows were not processed if modeless Edit(), Select() or Pick() was done from JavaScript. Now fixed.

- **Case 24528**

The exists property could not be set to true for *DATABASE ASCII cards. e.g. for model m, m.database.nodout.exists = true did not work. Now fixed.

- **Case 24519**

The select property did not get updated as a combobox widget selection changed. Now fixed.

- **Case 24483**

Toggling strict mode in the JavaScript debugger could cause PRIMER to crash when the script was run. This has been fixed.

- **Case 24029**

Deleting a model in JavaScript using Model.Delete() would make a modal JavaScript window become non-modal. Now fixed.

- **Case 23810**

If Widget.onTimer was used to hide a window as part of the callback then the script did not continue until the mouse was moved. Now fixed.

- **Case 23697**

It was not possible to obtain a label using the Set class function LastFreeLabel() in JavaScript if you wanted a label for a thick shell set, and there were no thick shell sets already in the model. This has been corrected.

- **Case 23678**

The JavaScript function FindSolidInBox() sometimes did not find all solids in the defined box. This has been corrected.

- **Case 23424**

If hover text was defined for a widget and '\n' was used to split the text into lines then it would cause problems if any of the lines were longer than 256 characters.

- **Case 23113**

Using try/catch did not work properly with the PRIMER JS API. It now does.

- **Case 21723**

A script could crash PRIMER if an object referred to an item in a model which had been deleted. Now the script will terminate with an error.

Selection

- **Case 25912**

Incorrect text could be displayed on mesh panel drop down buttons. This has been corrected.

- **Case 25081**

The following sequence of operations would cause (harmless) 'BadMatch' X11 errors on older Linux platforms, but could lock up PRIMER on more recent ones:

PART, Modify (or any other keyword)

Filter in object menu

Typing enough characters into the ascii text filter box to overflow it.

This was due to subtle errors in the GUI leading to a 'bad match' of windows on X11 windowing systems, ie Linux. It would not affect Windows platforms. Now fixed.

- **Case 24893**

When an object menu shows a single type of *ELEMENT, eg shells, for multiple models the an error in selection could occur in the following situation:

Menu shows results for this element type in more than one model.

One or more models contain > 10,000 instances of that element type.

One or more models contain < 10,000 instances of that element type.

In this situation all menu rows for all models will show a range of elements using [start .. end] syntax, and clicking on that row should select all elements in that start .. end range.

However if you clicked on the row of a smaller model with < 10,000 elements of that type you would only get the last element in that range (ie 'end') and not the full start .. end range.

The problem was that the syntax used for display was, to be consistent, always [start .. end] mode if any model contained > 10,000 items; but the parsing of a given row considered only that model. So if the model of that row had < 10,000 items the parser believed that it would be showing an explicit range of elements and picked only the last.

This has been corrected by making object menus 'remember' whether they are using explicit rows of elements of [start .. end] ranges, so that when a row is parsed it is treated correctly.

Sets

- **Case 25986**

PRIMER occasionally did not calculate the contents of _COLLECT sets correctly. Now fixed.

Shortcut

- **Case 24553**

Using the 'c' shortcut when the airbag reference coordinates and normal coordinates were swapped would swap the coordinates back again. Now fixed.

Sketch

- **Case 25038**

Sketching when moving the mouse over the clipboard row in an object menu could be very slow if the clipboard had a very large number of items in it. It is now significantly faster.

- **Case 22182**

If the sketch item was both blanked and off screen, the behaviour of single item (crosshair) sketch was not consistent with multi-item sketch in that a subsequent AC operation failed to draw the item into view space. Now fixed.

Text

- **Case 25500**

Using the 'text edit' function to edit Model comments or to text edit lines from the keyword editor could leave the temporary scratch file on the system.

This would only happen if the contents were unchanged, it was deleted correctly if the contents were modified. This was annoying, in that it could lead to a build-up of temporary scratch files, but did not stop PRIMER working correctly.

This is now fixed: these temporary files are now deleted regardless of whether they were modified or not.

- **Case 25297**

When using the Gvim editor on Linux as the editor for 'text edit' operations in PRIMER it is necessary to start it in 'no fork' mode so that it does not start a child editing process and then terminate the original launching process.

This is because PRIMER monitors the original process and takes its termination as meaning 'editing has finished' whereupon it inspects the file for any changes. Since the originating process terminates as soon as the separate editor is launched this means that text editing fails.

The fix is to launch Gvim using the '--nofork' argument, meaning that the PRIMER text editor preference should be set to something like:

primer*text_editor: /usr/bin/gvim --nofork

However this did not work because of errors in the way the text editor launched processes, it got confused by the extra argument.

This is now fixed: text editor strings can contain further arguments and these will not prevent their use so long as the original executable name, here /usr/bin/gvim, is a valid file on the system.

Timestep

- **Case 25430**

PRIMER did not calculate a modulus for MAT 57 correctly (for timestep calculation) if LCID was a table. Now fixed.

- **Case 24943**

Timestep for cohesive MAT184 elements is now calculated.

- **Case 24469**

Timestep calculation for MAT184 solids (cohesive elements) has been corrected.

- **Case 24336**

Timestep for MAT_169 when option extra = 3 has been corrected to include the effect of stiffness factors on the material card.

- **Case 23482**

Timestep of solids of formulation 16 was not calculated. Now fixed.

- **Case 23391**

wrong time step for MAT_LOW_DENSITY_FOAM - Fixed now.

- **Case 19523**

Timestep for beams which use _OFFSET was not calculated. The option may effectively shorten the length of the beam and so decrease its timestep. Now fixed.

Undo

- **Case 24631**

When splitting shells PRIMER could crash if some of the shells were in sets and the sets had comments stored in them. Now fixed.

- **Case 21867**

'Undo' for node-based operations such as Drag and Replace, in conjuction with *AIRBAG_REFERENCE_GEOMETRY, was producing incorrect results. This has now been fixed.

Units

- **Case 25457**

Units conversion would fail for *BOUNDARY_RADIATION. This has been corrected.

- **Case 24632**

Units conversion of ln(strain_rate) was not being handled. Fixed this.

- **Case 24550**

Fixed: MAT_FABRIC TSRFAC (when -ve) x-axis units of curve should be Time (previously no units used).

- **Case 23871**

Unit conversion for LOAD_RIGID_BODY has been corrected to work with moments/rotation.

- **Case 23423**

The units for load curve LCID on *FRQUENCY_DOMAIN_RANDOM_VIBRATION optional card 6 were the wrong way round. Now fixed.

User interface

- **Case 25772**

A child Javascript window that has the 'on top' property set failed to show the popup menu associated with the top left [-] options button. It would flicker and disappear.

This was because that menu was not inheriting the 'on top' status of its parent window correctly, and thus getting demoted to lie behind it. Now fixed.

- **Case 25439**

In the *DEFINE_CURVE_SMOOTH editing panel it was possible to curtail the options shown in teh curve type radio button set by moving the bottom border of the window upwards to leave a very shallow region for the data entry buttons.

This was the that height, rather than the true available height in upper region where it was drawn, was being used to determine the space available for that radio button. Now fixed by using the correct value.

- **Case 25104**

PRIMER will now automatically start edit panels in modify mode if there is at least one item in the active model(s) [previously it was if there was more than 1 item in all models].

Additionally, if there is only one item in the active model(s) PRIMER will no longer automatically edit it.

- **Case 24501**

The interactive build panel has been resized and the layout changed to improve readability.

- **Case 23972**

When both HISTORY_NODE and HISTORY_NODE_LOCAL cards are present, the correct number of DATABASE_HISTORY_NODE cards is now reported in the menu.

- **Case 23960**

The *CONSTRAINED_JOINT editing panel has been redesigned to make it smaller and less confusing.

- **Case 23771**

All filename and pathname handling has been upgraded to handle strings of length 512 characters.

- **Case 23370**

In models with very many include files, the drop-down from model checking which enables filter by include file can run out of space. The function which assigns space to drop-downs has been improved to allocate space more cleverly.

Write

- **Case 25870**

PRIMER is capable of writing a 'd3plot' (or PTF) file of a model suitable for reading into post-processors. The size of this file was limited to 2GBytes, equivalent to roughly 50m elements.

This limit is now lifted and the file can be any size. In addition if the model to be written contains labels greater than the 32 bit integer limit the file will be written in 64 bit format.

- **Case 24686**

If there are two or more models in the database then Model, Write requires selection of the model to be output. This selection can be performed either by selection from a menu or by typing in a model number, and it works fine.

However if a selection is made and then the main PRIMER window is resized the currently selected model is lost. This is an inconvenience when running interactively, but can be a disaster when replaying a macro which assumes a current model has been selected.

The error was that a change in overall window size requires the Main (model) panels to be redrawn, and this redrawing process is also performed when the number of models in the

database changes. The two operations were considered to be identical, and assumed that the number of models had changed, thus cancelling any current selection.

The two reasons for redrawing have now been separated, and resizing the PRIMER window will no longer cancel an existing model number selection in the main Model panel.

2.2 D3PLOT

2.2.1 Bugs Fixed in 12

Contour

- **Case 24489**

The criterion (direction) plot of 2D principal stress was being displayed in the wrong direction for a uniaxial shell test model in which the stress was exactly aligned with the element local axes.

The problem was a wrongly written ill-conditioning check which would only be triggered by this exact set of circumstances, ie pure compression with exactly no shear.

The calculation has been fixed and will now deal with that condition correctly.

Cut Sections

- **Cases 12634 and 24276**

When multiple models exist in a window and a cut section is active the user would expect that screen dragging the section would result in the sections cut through each model remaining co-planar (in deformed space).

This was true for drag translation, but for drag rotation the sections could sometimes end up no longer being co-planar. This was due to ill-conditioning in the calculation that turned a cursor drag motion into a rotation which could give subtly different results for the different models in the window, and build-up of these differences would quickly result in different plane orientations.

Drag rotation of a cut plane through multiple models in a window now imposes exactly the same rotation on the section in all models, solving this problem.

Data Access

- **Case 24649**

The data component 'Strain Rate' now appears with the rest of the strain based data components. For backwards compatibility it also still appears under the miscellaneous data components but it has been made more obvious that it applies to thin shells as well as solids and thick shells.

Envelope

- **Case 22370**

Historically an attempt to 'capture' in REPORTER a D3PLOT image that shows an envelope plot has failed. Either it gave no results or - worse - results that were wrong.

There were two reasons for this:

(1) The actions required to initiate an envelope plot in D3PLOT during playback of the REPORTER template were missing, meaning that the plot was not set up correctly.

(2) It was also possible to select a subset of states during 'Capture' of the image, but this list of states was not stored and so was lost during playback.

These two problems have been fixed as follows:

(1) The necessary actions to initiate an Envelope plot now take place, and it is set up correctly.

(2) In the REPORTER playback case this plot will always span all states in the model.

This 2nd point, using all states, is what most users do most of the time, so it should be satisfactory. However if users try to define a subset of states in the REPORTER 'capture' phase this will be intercepted and changes to all states, with a warning message giving an explanation.

Graphics

- **Case 26107**

Capturing a screen image in REPORTER from D3PLOT when running in 'batch mode' (which means no screen window is shown) resulted in failure on Linux and a crash of D3PLOT.

The crash occurred because D3PLOT was attempting to use hardware accelerated graphics when not running in a physical window, ie when the graphics card was not not properly 'connected to' the display. It has been fixed by reverting to software shading when in batch mode.

- **Case 25581**

When Shift Deformed was used in conjunction with Magnify Displacements the wireframe overlay on the plot could sometimes 'lag behind' when shift deformed was turned on/off, and not be drawn in the correct position until an explicit redraw command was issued.

- **Cases 25581 and 21652**

When an image is captured containing Airbag Particles, SPH elements or DES elements the elements are now automatically drawn as spheres in the image if the on screen symbol has been set to point.

This change has been made as although 'points' are quicker to draw they don't always appear the correct size in captured images due to limitations in the maximum point size supported on some machines when drawing to an image.

Using spheres is slower (this doesn't matter as they are only used when capturing the image) but the symbols end up the correct size.

- **Case 25428**

Predictive picking of Spring and Seatbelt Parts in D3PLOT would incorrectly put the label at the coordinate 0,0,0.

Version 12 has been modified so if the user is picking PARTS and the element selected is either a spring or seatbelt the label is now drawn at the element centre.

- **Case 25013**

When capturing an image (static or movie) any restraint symbols, ie SPCs, would fail to appear in the image file despite being visible on the screen.

- **Case 24913**

If you used the up and down arrow keys to cycle between models and then changed the drawing mode from SH to LI, then all the models changed to LI only if M1 was drawn. If a different model was being drawn the drawing mode got reset the next time you pressed up or down to cycle to another model. This has been fixed.

- **Case 24526**

In version 11 the Overlay mode for SPH and ABP elements could be set through the 'Display Options' menu but the Quick-Pick Overlay option didn't work for these element types.

- **Case 24462**

If the mesh overlay was changed from 'Free Edges' to 'All Edges', the overlay updated on all the solid, beam, shell and thick shell elements but not on SPH elements.

- **Case 24284**

If velocity arrows were drawn on SPH elements they would not be seen if the length of the velocity arrow was less than the radius of the SPH element. In version 12 the velocity arrows are now drawn with their tail positioned at the surface of the SPH element instead of the element centre so they are not lost inside the element.

- **Case 24282**

If the option to re-read models was used when multiple models had been loaded into D3PLOT then any window based properties such as SPH radius, beam plotting mode etc which had been set would not be applied correctly to the model after the re-read operation had completed.

- **Case 23952**

When using software computed normals (not the default on modern machines), the lighting in a model in a second window could occasionally go wrong being both 'the wrong shade' and also often too bright.

- **Case 23877**

If data values were displayed using dynamic labelling on spotweld solids then the data value would be displayed on all the visible faces which made it very difficult to read unless you zoomed right in on each spotweld.

Version 12 had been changed so that the data value is now displayed on just one face of the spotweld solids.

- **Case 23357**

Part-based elements would sometimes fail to draw in a state if the following sequence of operations was followed:

- Read model, turn off 'entity' switch for some category, eg solid.
- Close model.
- Reopen model, or open a new model with similar content.

At this stage no solids will be drawn because the entity switch for them is off.

Turn the entity switch on and the solids fail to appear in that state, although they will appear when the next state is rendered.

The problem was that part-based elements such as solids are subject to a 'coarse clipping' test which means that they will only be drawn if the part which contains them is on the screen. This test is based on the bounding box round the part, and this was being set up wrongly if the entity switch for that part's element type was turned off.

The same problem of 'started new model with entity switch off' could also be provoked in other ways, giving similar problems.

This has been fixed by making sure that when a part is first rendered within a state there is an 'is this the first time I have drawn you?' test which calculates the bounding box properly.

- **Case 23022**

If a ZTF file is present D3PLOT can obtain restraints and constraints on nodes, and display these if the appropriate switches are turned on in the Entity panel.

However this has caused some confusion since the symbols used for both REStraint and CONstraint are the same, making it difficult to tell one from the other if more than one category is turned on.

To compound the problem one class of constraint is 'Node on a rigid part', and if this is selected then all degrees of freedom are shown as constrained (which is correct), but this can be confused with the actual restraints - if any - on the rigid part.

So an extra category 'Restraint on rigid part' has been added which will show restraints in the appropriate degrees of freedom on nodes on rigid parts which have been restrained on their *MAT card.

Groups

- **Case 25953**

If D3PLOT read a GROUP .grp file written from an older version of D3PLOT and then you used the blank menu to blank groups, D3PLOT could crash.

Image

- **Case 21985**

If an image was captured containing multiple windows with envelope plotting turned on then the first image captured for a data component could have blank images where some or all of the windows should have been. If a 2nd image was captured without changing component then it contained all the windows.

Labels

- **Case 25926**

In the entity panel the 'Attached Nodes' switch for displaying node names did not work.

Measure

- **Case 25854**

Deleting the current measurement was not removing it from the graphics window. Pressing SH or moving the model updated the display. It now updates straight away.

The same problem has been fixed with the Delete All button.

- **Case 25023**

Measure node-to-node would return angles as NaN if the nodes were coincident. They are now returned as 0.0.

Menus

- **Case 25949**

When using the D3PLOT->T/HIS link the position of T/HIS graphs would get reset if the user minimised and maximised the main application window or iconised it.

In a standalone D3PLOT or T/HIS session manually resizing the application window by dragging out it's corners would also reset the position of any graphics windows that the user had manually moved but not any that he had manually resized.

- **Case 25942**

If the mouse scroll wheel was used while the mouse was either over the border of a window or over the green bar that divides the graphics area from the right hand menus the program could stop responding.

- **Case 22300**

If the option to see the size of the graphics windows was used (UTILITIES->GRAPHICS) then the new sizes were not applied to windows that were not on the current page even if the window tabs were selected for those windows.

- **Case 14239**

If D3PLOT had multiple windows open and the Wn selector had been used in one of the graphics windows to deselect that window in all the menu panels then the window tabs in the graphics windows would get out of sync with the ones on the menu panels if the user created a new window.

Movies

- **Case 25414**

If a movie was captured when you had a movie or static image in the background of the D3PLOT window and the T/HIS link was also being used then the model in the movie would appear to jump from one location to another after a few frames. This wouldn't happen on screen - just in the captured movie. This has been fixed in version 12.0.

- **Case 24668**

If a background movie was loaded in a Window in D3PLOT and a settings file was written out the movie would be loaded correctly when the settings file was replayed in a new D3PLOT session but the display of the movie background was sometimes not turned on.

If a number of points had been selected to align the model with the movie the points were not written out to the settings file and would be lost when the settings file was reloaded. In version 12 the point details are now stored in the settings file and can be edited after reloading.

Multiple Models

- **Case 24687**

If a reference node was setup in model 1 before a 2nd model was read in the reference node would not be applied correctly when values were contoured. Instead of displaying the relative values the absolute values would be displayed. Turning the reference node option off and on again would fix this problem.

- **Case 24675**

If a second model was read into Window 1 in D3PLOT then sometimes the model would be added correctly to the window but it's visibility switch (Window->Edit Windows->W1) would not be set correctly. This would often happen if the 2nd model contained a properties file.

Part Tree

- **Case 25011**

If the option to list contact surfaces in the part tree was turned on then two times as many rows would be displayed as there were contact surfaces in the model. These extra rows would be labelled with '0' and you couldn't do anything with them, but they shouldn't have been listed.

Properties

- **Case 24944**

If the option to reload a properties file into D3PLOT was used then after selecting a properties file for a model the tick box option to read the properties file wasn't automatically selected. This meant that if after selecting the file the user just pressed 'Apply' the file wasn't actually read.

Quick Pick

- **Case 25065**

If the Quick-Pick Find option was used while a model was loaded into different windows the cross-hairs highlighting the item were only drawn in the 1st window.

- **Case 25052**

If the Quick-Pick Find option was used while multiple models were loaded into different windows the location of the cross-hairs highlighting the item didn't update correctly if the magnification factor for displacements was changed. The cross-hair was located correctly in any windows containing the 1st model but not in windows for other models.

Read

- **Case 25876**

If multiple models were read into D3PLOT and D3PLOT was unable to read the LSDA/binout files for one of the models because it was still running then D3PLOT could crash if the user deleted all the models in D3PLOT.

- **Case 25092**

D3PLOT V11 will only read individual files up to 2GBytes in size (single precision).

D3PLOT V12 will read up to 8GBytes (single precision) or 16GBytes (double precision), these limits being determined by internal addressing issues.

These limits will be removed in future releases.

- **Case 24461**

Case 24461If a model was opened and the factor used to scale the radius of SPH elements was modified the new factor wasn't applied to a second model when it was opened.

Scripting

- **Case 25389**

The maximum number of widgets that could be made in a window was limited to 1000. This can now be changed by using Options.max_widgets.

- **Case 25064**

The Blank/Unblank, IsBlanked and IsDeleted functions crashed D3PLOT if you tried to use them with nodes.

- **Case 24256**

The following 2 bugs in the JavaScript function GetData() function have been fixed:

  - It always returned zero for segment contact forces.
  - It gave an error saying that the SURF item type was not valid.

An extra optional argument has been added to the GetData() function to switch whether blanking should be considered or not when calculating results (default is not to consider blanking). This is relevant to nodal contact forces as by default this function will return the sum of all forces at a given node for all surfaces using that node. Users can now blank all but the contact surface(s) of interest and turn this option on to restrict the results to the contact surface(s) they want.

Two new functions have been added to help working with contact surfaces.

GetSegmsInSurface() returns the start and end indices of the slave and master segments in a given contact. This can be used to loop over the segments.

SpoolNodesInSurface() can be used to spool through the nodes on the slave and master sides of a contact.

Settings File

- **Case 24676**

If a settings file was written out while the D3PLOT->T/HIS link was active the model index written to the settings file for any D3PLOT windows that were created after the T/HIS link had been started were wrong. This could cause a crash when the settings file was reloaded in another session.

T/HIS Link

- **Case 25306**

When the D3PLOT->T/HIS link was used with a model containing X-Sections a local triad was drawn on the X-Sections by default using a length of 1.0 for the triad axis as the correct length wasn't being calculated. This didn't matter if your model was in mm but if the model dimensions were in m then drawing the triad with a length of 1.0 resulted in a huge triad for each X-section.

D3PLOT 12.0 now only draws a triad if the user has turned on the option in the 'Display Options->X Section Symbols' menu and the triad size is calculated correctly.

- **24537**

When using the 'Locate' option in the D3PLOT->T/HIS link it wasn't obvious to users how to remove the locate symbols or to clear the items they had sketched. The 'Locate' popup menu now contains a 'Clear All' option that removed any sketched items or locate symbols.

User Interface

- **Case 22773**

In version 12 the maximum length for filenames (including the directory) has been increased from 256 to 512 characters

NOTE : Internally Windows supports filenames of up to 32768 but currently it does not support pathnames longer than 260 characters within Windows Explorer or the command line interface..

User Defined Data Components

- **Case 24269**

D3PLOT 11 could crash if user defined data components were created in a model contains SPH elements.

X-sections

- **Case 26083**

Selecting a LoadPath component could crash D3PLOT if the d3plot.components file had not been read or did not contain data to set up X-Section components. These are now made unavailable for selection if this is the case.

2.3 T/HIS

2.3.1 Bugs Fixed in 12

Curve Table

- **Case 25839**

If the curve table was set to 'Group Common Items' then the Style column was correctly shown for each row using the description 'Mixed'. The component, model and directory columns however could show incorrect descriptions.

- **Case 25838**

If columns were turned off in the curve table using the curve table view menu then the curve table wasn't updated correctly until the view menu was dismissed.

- **Case 24976**

If the display of columns in the curve table were turned on or off the positions of the buttons and scroll bars used to select which curves were displayed on each graph did not update immediately.

If the curve table was resized or dismissed and then re-displayed then the positions updated correctly.

Filtering

- **Case 25542**

If the auto regularisation switch was on any operations that used it would always take the value specified to regularise the curve. This was wrong if the curve had a smaller timestep than the one specified as you would effectively be filtering the curve. Any operations that use the auto regularisation now use the smaller of the two values.

The curve label now also includes the regularisation timestep used, e.g. if you use the C60 filter with regularisation on the label is now 'Node 1 (Reg 0.0001) (C60)', rather than 'Node 1 (C60)'.

General

- **Case 21840**

In version 11 of T/HIS it was not possible to turn off the display of individual injury criteria if more than one injury criteria had been calculated for the same curve.

Version 12 has been modified so that the curve properties menu now allows the display of each injury criteria to be turned on and off separately in both the graph area and the legend.

Groups

- **Case 24257**

If T/HIS read in an ASCII file containing PART GROUP definitions the groups would not be created correctly if they were not defined in order of increasing group ID in the file.

In version 12 the PART GROUPS can be defined in any order.

Menus

- **Case 24482**

In version 11 of T/HIS scrolling the list of curves in the curve manager using the mouse scroll wheel was very slow on LINUX if shaded buttons were being used. Turning off shaded buttons by setting the button graduation to zero (Top Tool Bar : Options->Menu Attributes) fixed the problem.

In version 12 the scrolling speed has been improved but it is still slower on LINUX if shaded buttons are used compared to flat buttons.

Operations

- **Case 23693**

If the automatic regularisation option was used with the FFT or IFFT function and the time interval was very small compared to the length of the curve then the FFT function could take a long time to complete. T/HIS now checks to see if the STOP button has been pressed within the FFT function and aborts the curve operation if it has.

Quick Pick

- **Case 22292**

When using the D3PLOT->T/HIS link the T/HIS Quick Pick option didn't always work correctly after doing something where the user could pick curves interactively (Operate Menu etc). After using the Operate menu then using the left mouse button for a Quick Pick option didn't work while using the right mouse to apply the Quick Pick option a group of curves did work.

Read

- **Case 25842**

If multiple models were selected to be read and one of them had already been read into T/HIS then the 'Extract curves to match model' option didn't work.

- **Case 25025**

In version 11 the option to search directories recursively for results only reported the 1st model it found in each directory of the directory contained multiple runs.

Version 12 now find and displays multiple models in the same directory.

- **Case 24652**

T/HIS incorrectly offered x,y and z moments as available data components for nodal velocity boundary conditions and not for rigid body velocity boundary conditions where they should have been available if data was read from the ASCII BNDOUT file.

If data was read from the BINOUT (LSDA) file then the correct components were displayed but if a FAST-TCF script was generated then it would incorrectly try and read moment data from nodes instead of rigid bodies.

Scripting

- **Case 23404**

T/HIS could get stuck in a loop if a JavaScript was run and T/HIS and a dialogue input command was used to try and read in a set of results that didn't actually exist.

```
DialogueInput('/MODEL READ', 'D:\\veh_21.key');
```

This has been fixed in version 12 but a workaround in version 11 is to use this instead:

```
DialogueInput('/MODEL READ', 'D:\\veh_21.key', '/');
```

T/HIS link

- **Case 25929**

If a D3PLOT model was closed while using the D3PLOT->T/HIS link and then reopened the 'Pick Visible' option for picking entities interactively in the T/HIS read menu would not be available.

- **Case 25070**

When running the D3PLOT->T/HIS link T/HIS could crash if a model was read into just T/HIS and the user selected the option to automatically extract curves to match some already read in from another model.

- **Case 25014**

If lots of curves were transferred from the XY-PLOT option in D3PLOT to T/HIS and the number of curves in T/HIS went past 1000 then the '+' and '-' buttons in the T/HIS curve manger that are used to map curves in chunks of 1000 at a time were not activated so you couldn't get to any curves past 1000.

(A work round in 11.0/11.1 was to create a single curve using any of the curve operations as this would then update the Curve Manager correctly).

- **Case 24384**

If when running the D3PLOT->T/HIS link the option to undock the T/HIS menus (Curve Manager, Operate, Maths, Seismic, Automotive) was used then the undocked menus would initially be the same width as the docked ones. This was different to the standalone version of T/HIS where undocked menus are initially twice the width of the docked ones.

In version 12 the undocked menus in the linked version are also now twice the width of the docked ones when they are initially undocked.

- **Case 23486**

If the directory selector was used inside T/HIS to select a directory to search for curve files in when using the D3PLOT->T/HIS link then both T/HIS and D3PLOT would lockup and stop responding to the user.

User Interface

- **Case 25845**

The default file filter in T/HIS has been changed from '*.*' to '*' when browsing for results files.

On a PC '*.*' showed everything including files without extensions so you could easily select 'binout' if you wanted to. On LINUX the file selector would only show files with extensions if the filter was '*.*' and this made it difficult to select 'binout'.

'*' works fine on both and shows all the files.

2.4 REPORTER

2.4.1 Bugs Fixed in 12

Capture

- **Case 24933**

In the D3PLOT object command file, extra commands longer than 80 characters did not work. This has been fixed.

Data access

- **Case 25630**

The Job File option for T/HIS objects can now be left blank. This is useful if you want to read results from curve files, csv file or the keyboard rather than an LS-DYNA analysis. Previously REPORTER would not allow this.

Fasttcf

- **Case 24979**

FAST-TCF objects did not use the job filename when generating. FAST-TCF would look in the directory for the job. If there were multiple jobs in the same directory it could choose the wrong file. This has been fixed.

Graphics

- **Case 25848**

Circles and ellipses were not rendered correctly in pdf files. Now fixed.

Image

- **Case 24988**

REPORTER could crash if a library image was inserted very near the top or right hand edge of a page. This has been fixed.

Pedestrian

- **Case 24491**

The library scripts for contouring the bonnet with HIC values, 'pedestrian_zone_area_from_csv.js' and 'pedestrian_zone_area_from_variables.js', could be slow if they had to extrapolate a lot of values, e.g. when there were no impact points on the bonnet perimeter. This is now much quicker.

2.5 Shell

2.5.1 Bugs fixed in 12

General

- **Case 24278**

If a list file was used to submit a number of jobs then the Shell was ignoring the memory to use for each job if it was specified in the keyword file with '*KEYWORD xxxxxx'. Now fixed.

User Interface

- **Case 242277**

The Dyna submission window and program option windows did not have an [X] button in the top right hand corner to close them.

3 Enhancements

The enhancements and bug-fixes for each program have been broken down into a series of topics. Within each topic enhancements and bug-fixes are listed by case ID (most recent first).

3.1 PRIMER

3.1.1 Enhancements in Version 12

Adhesive

- **Case 23804**

A new adhesive type has been added allowing the user to create 'patches' of adhesive based on a source mesh.

Airbag

- **Case 13579**

When a *AIRBAG_HYBRID(_JETTING(_CM)) definition is converted to type *AIRBAG_PARTICLE PRIMER now attempts an 'intelligent' conversion between the two definitions.

In summary set ids, loadcurves, gas information and jetting (if present) are transferred from old to new definitions. The transfer is a long way from complete because the two airbag types are very different in detail, but it preserves a reasonable amount of data.

A warning message is printed when the conversion is performed explaining what was done in more detail.

Animation

- **Case 22072**

Animation of BOUNDARY_PRESCRIBED_MOTION definitions should now be faster, particularly for relatively simple definitions and for fine meshes.

- **Case 21617**

The animation function for BOUNDARY_PRESCRIBED_MOTION and BOUNDARY_PRESCRIBED_FINAL_GEOMETRY now includes repeat/play once options

Belts

- **Case 24899**

The following changes and enhancements have been made to the seatbelt fitter:

+ The distinction between a 'Pelvis' (free) slipring and a 'B-Post' (limited rotation axis) one has been made explicit.

Existing decks which just have a slipring defined in the belt path will work as before, but passage through the belt fitter will update them to mark the slipring as the appropriate type.

Making a slipring explicitly 'free' gets rid of the problems that users have encountered previously when trying to make belts with geometry other than 'retractor and slipring on b-post', since it will not make any assumptions about geometry or belt twist.

+ The definition of an 'acute' angle in the belt path is now user-definable.

By default an angle tighter (less than) 90 degrees is still considered to be acute, and will imply a new belt segment. However this angle can now be changed so, for example, making it smaller will allow belts to have tight loops of continuous mesh.

+ Transverse friction of the belt is now user-definable.

Previously belts had a built-in and fixed transverse friction coefficient (of approximately 0.2) that resisted sliding across a dummy once in contact.

This default is now zero, meaning that belts are more likely to slide and therefore to adopt the 'shortest distance between fixed points' path, but it can be defined as any value in the range 0.0 to 1.0 by the user.

+ Reduced transverse friction also means that the belt is more likely to 'fold back on itself', especially in the lap region. Therefore the default transverse limiting curvature angle has been changed from zero (unconstrained) to 30 degrees, effectively giving the belt path a little bit of transverse bending stiffness.

+ Default #iterations before assuming convergence has been increased.

In order to give a belt more time to slide across a dummy the default maximum number of iterations in both manual and auto-refit processes has been increased. If convergence occurs then fitting will cease as before, but if it does not the larger number of iterations that are now the default make it more likely that the 'shortest path' will be found.

+ Self-contact of the belt is now considered.

Historically the belt fitter calculated contact between belt and dummy, but not between overlapping sections of belt such as those close to the pelvis slipring.

When 1d belt elements were used close to sliprings this did not matter, but now that 2d belt elements are becoming commonplace this could give rise to penetrations if the belt was in a single-sided contact.

Overlapping sections of belt will now keep themselves apart, with the 'later' (higher numbered path points) section being further out from the dummy.

+ The titles of cross-sections added to a belt have been improved.

These have always had labels and titles giving their position on the belt path, but for cross-sections next to a retractor or slipring the title now gives the adjacent retractor or slipring number as well as the distance from it. This should make it easier to identify which cross sections should be used for which elements.

+ The overall length of the belt is now reported, and can also be assigned to a parameter.

Previously the length of each belt segment was reported as it was meshed, but the overall belt length (ie the sum of these) was not calculated. This sum is now reported as part of the post-meshing summary.

In addition a new ability to assign this length to a parameter has been added. This works as follows:

- The user designates a parameter (by name).

- This is a normal *PARAMETER definition, type real, that can be referred to anywhere in the model.

- When a belt is (re-)meshed the value of this parameter is updated to become the new total belt length, and any cards which refer to it are themselves also updated with the new value.

The intention is that this parameter should be used to define characteristics of belt-related items, for example seatbelt sensors, which depend on belt length and which can thus be updated automatically whenever the belt is remeshed.

+ The need to remember a segment set as part of the *BELT definition has been removed.

Internally PRIMER converts the elements and parts used to define the dummy into a segment set which is then used when calculated belt-to-dummy contact, and historically this segment set has been remembered as part of the belt definition.

There is no real need for this as the segment set is recalculated each time prior to fitting, and keeping it can present problems when a new dummy is swapped in because it may refer to nodes that no longer exist. Therefore this segment set, and the reference to it on the *BELT card, is no longer written out to a keyword file unless it is used explicitly in some other context - for example to define a *CONTACT.

+ Replace dummy has been updated and improved.

This feature was in PRIMER 11 but was difficult to use and thus of limited usefulness. It has been improved both functionally and ergonomically to the point where it should now be a useful aid to replacing dummies.

- **Case 24841**

In the seatbelt fitter the titles assigned automatically to database cross-sections now include the label of the adjacent retractor or slipring if they are at such points. This makes it easier to identify which cross-section belongs where during post-processing of data.

- **Case 24656**

When meshing a seatbelt PRIMER always reported the centreline length of each segment (torso, pelvis, etc) but not the overall length.

The total length, ie the sum of all segment lengths, is now reported as well.

- **Case 23000**

Some clients have complained that the belt fitter in PRIMER doesn't always give the 'correct' (whatever that is!) straight line fit across a dummy. What they mean is that it doesn't necessarily give the absolute shortest path; also that the final path can be very dependent on the initial points used to define the base path.

This is, at least in part, because the belt fitter has a pseudo friction model which limits movement at points where the belt is in contact with the dummy, emulating the effects of friction. In V11.x the effective coefficient is about 0.3, and this is hard-wired in with no means of changing it.

In V12 the following changes have been made:

- The friction coefficient now defaults to 0.0 giving a more 'slippery' fit that is more likely to give the shortest path, and will also be more likely to give a final shape that is not so dependent on the initial path point locations.

- The friction coefficient is now user-definable in the range 0.0 (slippery) to 1.0 (sticky). This can be changed interactively using fitting 'parameters', and is stored in the *BELT keyword definition.

Users wishing to revert to pre V12 behaviour should find that a friction coefficient of about 0.3 gives very similar results.

- **Case 22171**

A criticism of PRIMER's belt fitter is that it is hard to use when fitting a belt to a child seat. Several changes in V12 help with this:

- Explicit selection of slipring type avoids the unwanted twisting that could occur previously if PRIMER tried to fit a 'B post' slipring to an inappropriate location.

- Explicit control of belt orientation at slipring locations, especially by setting a vector from 2 nodes, makes it a lot easier to slot the belt accurately through gaps.

A worked example of fitting a belt to a child seat has been added to the user manual.

- **Case 22091**

The standard 'check' functions in PRIMER missed the following error:

Model with 2d seatbelt requires a node set EDGSET to be defined on the SECTION_SHELL card for the 2d belt elements. This set contained a node which was not on a 2d belt element which caused a failure to initialise in LS-DYNA.

A check for this has been added.

- **Case 21714**

There is a request to 'remove the restrictions on rotation of a B-Post slipring'.

Before V12 PRIMER automatically chose the slipring type (B-Post or Pelvis) from the adjacent belt geometry. From V12 onwards the user chooses between B-Post and 'free' sliprings, and the latter will give the freedom requested here.

Checking

- **Case 25452**

It is now possible to control whether or not master categories (eg Airbag, Dummy, etc) get checked during Model check via preferences in the oa_pref file.

This can speed up checking in large models by removing checks for things either known to be correct or which which will not affect an analysis.

- **Case 25311**

The speed of Model Check has been improved, and some of these improvements will also provide modest speed-ups in other PRIMER functions. A summary of the changes made is:

Internal 'flagging' functions, used heavily in model check but also generally, have been parallelised. This starts to show benefits for models with more than 1m elements, and becomes more significant as model size increases. At 20m elements it probably gives around a 20% speed up.

Contact checking, a heavy user of cpu time during model check, has had its existing parallelism further improved. Again, the benefits show up most clearly in larger models.

Some specific bottlenecks, for example nodal initial velocity and solid adhesive checking, have been recoded to run much faster with large models.

In a typical large model the improvement in speed of Model Check should be at least 20%, and where previously it was slowed down by particular bottlenecks that have been fixed the improvement will be better still.

- **Case 25147**

Joint coincidence checking (for N1N2, N3N4, etc pairs) has always been a bit of a problem because of a dimensionless tolerance of 1.0e-3 hard-wired into LS-DYNA.

From LS971 R7.1 onwards this 1.0e-3 check is now a warning rather than an error in LS-DYNA, and a new check (an error) of 1.0e-4 times the distance N1N3 has been added for non-spherical joints.

PRIMER handles this as follows:

If the output version is earlier than 971 R7.1 the 'old' logic applies, to wit:

- 1.0e-3 is a fixed dimension error

- 1.0e-6 x the model diagonal is a warning.

If the output version is 071 R7.1 onwards:

- 1.0e4 x the N1N3 distance is an error

- 1.0e-3 is a fixed dimension warning

- 1.0e-6 x the model diagonal is a warning.

In the post 971 R7.1 case the tests are applied in the order given here and, to cut down on unnecessary messages, if an error is detected then any subsequent warnings are not written. This is because failure of one of the criteria above is likely to fail the others too, and it doesn't really add to understanding of the problem to report it in multiple different ways!

- **Case 24290**

A new error check has been added for when and are both zero on the *ELEMENT_MASS_PART card.

- **Case 22175**

A warning check has been added if a node is referenced by a null (*MAT_NULL) shell/beam, and is not referenced by any anything else.

- **Case 22169**

A new optional warning check has been added to issue a warning if release flags are set on *NODAL_RIGID_BODY's. This is off by default.

Connections

- **Case 23802**

Support for reading VIP connection files has been added into PRIMER. Also, if the panels being joined is referenced by CAD name rather than CAE PID, PRIMER will look at the Part CAD name (from an imported BOM) or the Part heading to determine what to connect.

- **Case 23235**

Wildcards are now allowed when defining connection layers (Part ID type only) on the connections table. A '?' represents one digit, and a '*' represents any number of digits. So, a layer part ID of 10?? will reference any shell part in the model with a label between 1000 and 1099. A layer part ID of 10* will reference any shell part in the model with a label that begins with '10'.

- **Case 23027**

Up/down arrows will now scroll the selection on the connection table and if the mode is set to sketch or show weld and panels, that action will be automatically applied.

- **Case 22937**

In DEFINE_HEX_SPOTWELD_ASSEMBLY panel, added an object menu to select multiple solids at the same time and listed in ascending order in the list.

- **Case 22164**

A preference has been added for the user to specify valid spotweld material types (default is only MAT_SPOTWELD).

- **Case 21366**

Connections now store the name of the xml file from which they have been created. This information is displayed on the connection table under the header Xml Filename.

- **Case 21174**

A new option has been added to create connections using assemblies or part sets to define the parts joined together. When using this option, PRIMER will just connect to any shells within the specified group of parts in the vicinity of the connection point/line/patch.

- **Case 19629**

Master Connection Files (.mcf) can now be read and written by PRIMER.

Constraints

- **Case 22173**

New fast part pick and autocreate options have been added to the *CONSTRAINED_RIGID_BODY edit panel.

Contour

- **Case 24661**

Added 'Shell Normal/Align' button on Contour->Shell Normal->Setting panel. This will give direct access to Normal/Align menu

- **Case 24655**

The ability to colour the structure by material number (eg 20 for MAT_RIGID) has been added.

This is done by contouring: select SI/CT contour category Matl Props, component Material Number

Cut section

- **Case 20655**

The ability has been added to reverse the direction of a *DATABASE_CROSS_SECTION definition, or a number of definitions at the same time.

- **Case 15950**

PRIMER now considers *ELEMENT_SHELL offsets in the following contexts:

Cut-section graphical display, and also section properties display and calculation.

Normal graphics if the Display Options, shell true section switch is turned on to either flat or smooth. (This is not the default)

Contact penetration checks.

All of the above will consider offsets defined by NLOC on *SECTION_SHELL or *INTEGRATION_SHELL cards, also the _OFFSET suffix on the *ELEMENT_SHELL card.

Note that LS-DYNA, and hence also PRIMER, only consider offsets during the contact penetration calculation if:

Field CNTCO is set on *CONTROL_SHELL

The contact surface is not an AUTOMATIC type

In addition nodes on the slave sides of NODES_TO_SURFACE contacts do not 'inherit' the offset of their parent shell.

Notes:

1: This is nothing to do with the optional OFFSET suffix available for certain contact surface types.

2: For more information on how and where offsets are applied to contact surfaces see the LS-DYNA user manual remarks on data field CNTCO on card *CONTROL_SHELL.

3: Offsets during contact are not displayed by default in contact penetration plots, but if the 'as thick' option to show the thickness of contact segments is selected these will also show any offsets.

Database

- **Case 21778**

There have been several requests for PRIMER to return more memory to the operating system when it is no longer needed, thus reducing the process size. In particular people can be surprised that deleting a model does not always result in a corresponding reduction in the PRIMER process size as reported by the operating system (ie by Task Manager on Windows, or 'top' on Linux).

PRIMER does in fact have strategies for returning memory, but in order to be robust it will only do so if:

+ There are no models in memory.

and

+ All storage for a given data category (eg *NODE) has been returned to the memory allocator as 'no longer in use'.

The reason for this conservative approach is that if a piece of memory is accidentally referred to after it has been returned to the operating system then PRIMER will crash, and we prefer that not to happen!

If only a subset of models are deleted, leaving one or more behind, then PRIMER will not return that memory to the operating system, but rather will retain it for future use. This retention is efficient: memory released from (say) nodes in model A will be reused for nodes when model B is read in; therefore repeated read/delete cycles for models will not result in excessive build-up of memory use.

You *can* persuade PRIMER to reduce its memory usage, but to do this you will need to delete *all* models in the database. However even then memory will not reduce to the level of a new PRIMER session, so if memory usage on your machine is tight it would be better to close the PRIMER session altogether.

Deleting

- **Case 22184**

For Remove->cleanup, the user can now specify the cleanup and individual include, or PRIMER assembly, rather than the whole model.

- **Case 18630**

Cleanup Unused has historically only applied to a whole model, but this has been extended so that it can also be applied to:

Selected include files.

Selected part tree assemblies.

If either of the above are chosen then only items in the selected include files or assemblies will be eligible for cleanup. (The ability to clean up a whole model still remains.)

This capability is available from the Remove, Cleanup menu as before; also from the Part Tree (include files and assemblies), and from the Include tree (include files only).

Dummies

- **Case 25492**

A new option has been added to Dyna dummy positioning to remove rigid locks between assemblies. Sometimes rigid body constraints are created in the model across assemblies to simulate the locking of assemblies relative to each other during a crash test. During positioning these would not be locked so should be removed during the setup of the pre-simulation analysis model.

- **Case 25366**

Added a new option in dyna dummy positioning. When on, PRIMER will look for CONSTRAINED_RIGID_BODY definitions that span assemblies in the dummy and deletes them. This is to account for these rigid connections found some dummies that you would want in the full analysis, but not during positioning.

- **Case 23023**

'Soft' stop angles have been added to the dummy positioning tree file.

These are optional and provide a means of limiting articulation of joints to +ve and -ve angles that are less than the limiting 'hard' stop angles defined on the *CONSTRAINED_JOINT_STIFFNESS card. This can be useful when utilising the full 'hard' angle range results in initial penetrations since a smaller 'soft' angle can be set to limit articulation so as to avoid these penetrations.

'Soft' angles are set in PRIMER's dummy tree file, which is placed after *END, so the subsequent LS-DYNA analysis does not 'see' these values which only have significance during pre-processing.

Edit

- **Case 24270**

A new tool 'Text edit' has been added.

This is a generalised version of the function of the same name available from edit panels, but this will work on any keyword in PRIMER with a couple of exceptions listed below.

The keywords, which must already exist, are exported to a mini keyword file and an editor is opened on that file. The user can then edit this file manually, and can also add new entries to it.

When editing is complete the user closes the file and PRIMER reads it back in. Existing definitions read in this way supersede the originals, equivalent to normal PRIMER editing. New definitions are added to the model as if they had been read in normally from a keyword file.

The keywords that currently cannot be processed in this way are:

*INCLUDE and *INCLUDE_TRANSFORM

*CASE

The reason is that these are extremely complex to process, especially in conjunction, and the simple 'read and update' process is not sufficient to handle them.

- **Case 23795**

Historically the 'Copy Existing' button in an editing panel has only offered for selection other items of this type in the same model as the item being edited.

This restriction has now been relaxed, and items of this type in any model in the database can be copied.

Elements

- **Case 23349**

Added a 'Reverse' button in the Element->Beam menu to swap nodes N1 and N2 of the beam elements

- **Case 17293**

Added edit panel and checks for *ELEMENT_DIRECT_MATRIX_INPUT.

- **Case 14005**

Added the create/edit panel for the *Element_Direct_Matrix_Input card into PRIMER

Graphics

- **Case 25097**

SPH elements in PRIMER have historically been drawn as spheres of a fixed size (20 screen units).

Now if enough information is available to compute their true radius this is used instead. It requires the mass/volume field on the SPH element card to be non-zero, and if a mass is supplied then the material density must also be known.

If these data are not available the graphics reverts to the 'old' logic of a fixed radius sphere.

- **Case 23816**

The ability to plot 'true' beam cross-section shapes was added to PRIMER 11, but this did not take into account any offsets. This has now been added, and offsets defined via:

*ELEMENT_BEAM_OFFSET (W vectors)

*SECTION_BEAM (NSLOC and NTLOC)

*INTEGRATION_BEAM (SREF and TREF)

are now all considered when beam true section display is turned on.

This also means that when a cut-section is made through beams and the 'properties' calculation is invoked the beam sections will include any offsets.

- **Case 22969**

An option has been added to turn off the display of the numbers on the graticule.

- **Case 22186**

Options have been addded to the colour menu to set the colour for *CONSTRAINED types - NODAL_RIGID_BODY, RIGID_BODY and EXTRA_NODES. This will aid the user in telling these items apart, for example during a find attached operation.

- **Case 15722**

Added sketch function to following keywords:

*DEFINE_STAGED_CONSTRUCTION_PART

*DAMPING_FREQ_RANGE

Few minor changes in sketch related functions to enable 'SKETCH' for following keywords:

*BOU_PORE_FLUID (draws the part or set of parts)

*BOU_PWP_TABLE (draws the part or set of parts)

*BOU_PWP_NODE or SET - (draws the node or node set)

*INITIAL_STRESS_DEPTH (or SET)

*INITIAL_PWP_DEPTH (or SET)

*LOAD_GRAVITY_PART (or SET)

*LOAD_STIFFEN_PART (or SET)

- **Case 12774**

There is now a Display Option to draw shells using their true thickness.

The default is still a single facet that is infinitely thin on the nodal plane of the shell, now referred to as 'thin'; but two 'thick' options have been added:

Thick flat draws each shell as a separate block, with no attempt being made to join up adjacent edges.

Thick smoot as above, but attempts to make a continuous surface over adjacent shells by joining up their edges.

The two 'thick' modes take into account the actual thickness of the shell, including any variation across nodes (T1 to T4). They also take into account any offset defines via *ELEMENT_SHELL_OFFSET, or by one of the NLOC arguments on *PART_COMPOSITE or *SECTION_SHELL.

If a shell is a composite, ie *PART_COMPOSITE or *INTEGRATION_SHELL with multiple layers, lines are drawn on the 'side face' showing each layer, making it possible to check thicknesses.

Finally if shell capping is used during cut-section display then in 'true' mode the cut section will show the actual cut shape through the true 3d shape of the shell element.

Include

- **Case 23740**

PRIMER 12 now supports *CASE in combination with *INCLUDE_TRANSFORM.

PRIMER 11 did not support this (we ran out of time during development) but this restriction is now lifted, and both *INCLUDE_TRANSFORM and *DEFINE_TRANSFORMATION cards may now be used inside *CASE statements.

Other transformation keywords *PART_MOVE and *NODE_TRANSFORM are also now supported inside *CASE.

- **Case 15912**

Added an option to read include stamped part as a new model in plastic strain view mode. It allows the user to read the stamped part by right clicking over stamped part in include part tree.

- **Case 12069**

Added include file selection/viewing feature on database edit panels.

Keyword

- **Case 24545**

New popup option '1.0' is added for _GEOMETRY cases.

- **Case 21378**

*RAIL_ keyword editing panel now supports 'Sketch' functionality. In *RAIL_TRACK, it sketches two beam sets BSETID1 and BSETID2 and in *RAIL_TRAIN node set NSETID is sketched.

- **Case 20649**

The popup in the IMFLAG field in the *CONTROL_IMPLICIT_GENERAL edit panel can now be used to select loadcurves

- **Case 19808**

A shortcut 'K' button has been added to edit panels that opens the LS-DYNA keyword manual at the relevant page.

- **Case 18976**

Added hover text to fields on *DEFINE_CONNECTION_PROPERTIES keyword

Kwd editor

- **Case 20669**

The keyword editor allows you to highlight multiple rows, and a change to a data field in any highlighted row is propagated to the same field in all other highlighted rows.

However if you choose [right click, edit] to edit one of the highlighted items this launches the editing panel and edits it, but when the edit is finished any changes are historically not propagated to other highlighted rows.

This lack of capability has been fixed. After the edit is completed any changes between the pre-edited and post-edited definitions are found, and are propagated to other highlighted rows. Such changes are any combination of:

- Changed data fields

- Changed keyword suffices (which may en/disable/change data fields)

- Changed include file membership.

Since these could lead to many changes in the model they are listed first, and the user is required to choose whether or not to apply them.

Macro

- **Case 23181**

A new function MacroFunction() has been added to JavaScript which will return the equivalent macro commands for events such as button presses, popups etc.

Materials

- **Case 25487**

When using the material editor to change the type of a material, for example from MAT_NULL to MAT_ELASTIC, PRIMER will attempt to copy over data from the old material to the new where the name of the data field is the same. For example density RO in the old material will be copied to field RO in the new one, even if it is in a different location on the card.

This process depends on the names of data fields being the same so it fails where different names have been used for the same data on different material types. A typical example is Young's modulus which is 'E' on some materials but 'YM' on others, and hence does not get copied.

Therefore a list of alias names has been added which permit data fields to be copied if:

- Both old and new materials match one of the alias name pair

- The units of the two data fields are the same.

At present this only matches 'E' and 'YM' with the proviso that both must use the same Modulus units. Further pairs will be added in the future if and when they become apparent.

- **Case 22336**

Added the ability to filter selection of *MAT and *PART in object menus by material option _ADD. Currently the options are:

*MAT_ADD_AIRBAG_POROSITY_LEAKAGE

*MAT_ADD_EROSION

*MAT_ADD_PERMEABILITY

*MAT_ADD_PORE_AIR

(There are other _ADD suffixes under *MAT, but they refer to parts and not to materials.)

- **Case 21299**

Added pref for Max strain for curves with ETAN option with default 1.0.

- **Case 20654**

Both the scalar and the keyword editors for materials now offer drop-down menus to select restraint codes for the rigid material fields CMO, CON1 and CON2. This saves having to remember or look up the meanings of the various options.

- **Case 9873**

Drop-down menus have been added for the CMO, CON1 and CON2 fields on the MAT_20 (rigid) editor panels.

Mechanism

- **Case 16722**

The ability to move *DATABASE_CROSS_SECTIONs with Mechanism or Dummy assemblies during positioning has been added. The details are:

- Prior to positioning a scan of all *DATABASE_CROSS_SECTIONS is performed. Only _PLANE types with a finite part set PSID are considered.

- A cross section is associated with an assembly, and is moved with it if one or more parts within part set PSID have been used to define that assembly.

- However if a cross section contains parts that are used in more than one assembly it is not possible to tell which it should be associated with, so the section will not be moved. In this situation the user is given the option of 'cloning' the section into as many definitions are required to make them unique to all parts that the original spans, and if selected these sections *will* be moved with their respective assemblies.

- Parts that are shells using *MAT_NULL are ignored for the purposes of determining usage in assemblies, since null shell parts are often used both to coat dummies, hence spanning several assemblies. These parts contribute little or no force, and it would be annoying if they were treated as errors.

- Movement of sections with assemblies is switchable at the Mechanism / Dummy level, and the status of this switch (default = on) is stored in the keyout file.

Menus

- **Case 23673**

Visualisation of *BOUNDARY_PRESCRIBED_FINAL_GEOMETRY definitions can now be toggled on or off from the PRIMER Entities menu.

Mesh

- **Case 23712**

All meshing functions consolidated under one Mesh tools panel.

- **Case 20981**

Added 'create true thickness solid from shell' option to shell mesh extrusion.

- **Case 17796**

When remeshing an area or creating a hole PRIMER could create facing triangles in the mesh. The algorithm has been changed and this should happen less often.

Model build

- **Case 17712**

PRIMER now has a preference that will enable the writing of all includes (rather than just the master file) to build directories during a model build operation using csv files

Nodal rb

- **Case 23493**

Added a 'COMBINE' button in the main NRB(Nodal Rigid Body) docked panel to combines multiple NRBs into one. Parent nrbs are deleted and a panel will pop to select model in case of multiple models to combine NRBs.

The new NRB and new SET_NODE are created in layer (include file) obtained by following rule:

1. If all selected NRBs are in the same include, create the new NRB in the same include

2. If not, create the new NRB in the current layer.

3. Using the same logic for new SET creation.

Node

- **Case 17811**

A starting or seed node label can now be specified for those creation methods that could spawn multiple nodes - line, surface, geometry points, geometry curve. Previously, these labels would be auto-created.

- **Case 14498**

Added post end keyword *TARGET_MARKER with large label support

Orient

- **Case 23175**

The orient translate and rotate function is now available on the command line.

Output

- **Case 26114**

There is a special OEM version of LS-DYNA 971 R6.1 which looks at the data field W-MODE on *CONTROL_SHELL, something that is not normally written in standard 971 R6.1 format.

PRIMER will read this data field unconditionally, but if standard 971 R6.1 output format is selected it will not write it out, since the original version of LS-DYNA did not specify the required extra lines on *CONTROL_SHELL.

The preference 'allow_wmode_r61' has been added, and if this is set to TRUE then the extra lines and data field W-MODE will be written on *CONTROL_SHELL in 971R6.1 format.

- **Case 25341**

The existing Model Write d3plot/PTF file capability only write the basic control block and element topology of the model.

This has now been extended to give the option of writing 'state 1' of data, which will still be based on the undeformed nodal coordinates but will include:

- Shell thicknesses

- Any *INITIAL nodal velocities

- Any *INITIAL nodal temperatures

- Any *INITIAL stresses or strains for solids, shells and tshells.

This will also provide somewhere to display user-defined data, should users wish to use this file as a 'chassis' for displaying that.

- **Case 24541**

It has become a problem that the output formats of *BELT, *DUMMY and *MECHANISM in V12 contain extra cards and are thus not readable in earlier PRIMER versions. So a preference

primer*mdumm_keyout_version: v11 | v12 | current

has been added.

The default is 'current', but setting this to 'v11' will generate output in the PRIMER 11 format. A warning will be issued if any of the data fields omitted are non-zero.

Parameter

- **Case 24657**

Previously the top level parameter panel, which shows a summary of all parameters in the model, only displayed 10 rows regardless of the panel height.

This has now been amended and as many rows as will fit into the panel, up to a maximum of 100, will now be shown. Therefore making the panel taller will display more parameters.

In addition a column to show the include file in which the parameter definition lives has been added. This is (deliberately) narrow so a long filename will be truncated, but hovering over an entry will give a popup showing the full filename.

Part tree

- **Case 8772**

A red/green indication has been added to the part tree showing blanking status of parts/includes.

Pedestrian

- **Case 24931**

Hard point detection has been added to the pedestrian markup script.

The script can automatically detect and create impact points close to user selected hard parts. Alternatively points can be created manually by picking on the vehicle surface or on the hard parts.

A contour plot can also be drawn on the vehicle outer surface to indicate the proximity to the hard parts.

- **Case 24648**

An option has been added to the pedestrian markup script to create a circle of points around the automatically and manually created points. This can be used to do sensitivity studies.

- **Case 24641**

The pedestrian markup script can automatically calculate the angle, velocity and mass values for the upper leg impactor, but it assumed the model units were mm,kg,ms. An option has been added to use mm,T,s instead.

- **Case 23988**

For the GTR protocol the pedestrian markup script will now put impact points on the impact zone boundary lines when it generates a grid of points.

- **Case 23982**

A log file now gets written out by the pedestrian markup script when the csv file containing the impact points is written out.

The log file contains information on:

- The inputs used to create the lines

- The X,Y,Z coordinates of the impact points

- The velocity, target mass, bonnet height and bumper lead for upper leg impact points.

- **Case 23706**

The EuroNCAP v7.0 protocol (2014) has been added to the pedestrian markup script.

- **Case 22248**

An option has been added into the pedestrian markup script to label the impact points using the 'GM style'. This uses an 8 digit label that encodes information about the impact point. The default is still to use the method defined in the protocol.

- **Case 21927**

An option has been added to the pedestrian markup script to find head impact points that could potentially impact on to hard parts below the surface of the bonnet.

- **Case 21456**

The protocols state that if the bumper is identifiable the upper bumper reference line should only be marked on those parts. Previous versions of the pedestrian markup script did not allow for this option and the reference line would be marked on the parts selected for all the other reference lines (i.e. it assumed that the bumper was not identifiable).

A new button has been added to select bumper parts. This is optional and if nothing is selected then the line will be marked on the parts used for all the other reference lines (default).

- **Case 19154**

PRIMER's model-build-from-csv function now includes two new master style options:

- 'GM style' generates one user-defined file to which all *DEFINE_TRANSFORMATION definitions are written. Each *DEFINE_TRANSFORMATION definition is given a unique label

which is equivalent to the directory name, if valid. Each master model then refers to the appropriate *DEFINE_TRANSFORMATION by its label. Also, a couple of extra cancelling transformations are added such that the last translation represents the aim point. Additionally, a title string may be specified. The tile for each *DEFINE_TRANSFORMATION card would be a combination of the label, the aforementioned string, and the depenetration type.

A new 'First impact point only' option has been added which is specific to the GM style. This option will generate *DEFINE_TRANSFORMATION information (single file as above) for all loadcases but will only write out the first master model.

- 'CASE style' generates a single user-defined file to which all data is written. No additional master models are generated. Data for each loadcase - including *DEFINE_TRANSFORMATION - is specified under a distinct *CASE definition. *INCLUDE for the model and *INCLUDE_TRANSFORM for the impactor are only written once for the process.

- **Case 17126**

A new model-build-from-csv option has been added for the PEDHEAD build type that will permit target point projection along the line of flight. The old option of projecting vertically along Z is still available.

Also, 'Sketch' and 'Pick' in the interactive panel will now show the original point and the projected point.

Preferences

- **Case 17753**

The names of the preference files that are read are now written to the dialogue box when editing preferences

Read

- **Case 25493**

Recent files that have been read/written within PRIMER are now available via a dropdown on the file read input text box.

- **Case 17929**

In version 12 of PRIMER models can be opened on a PC by using drag and drop. PRIMER assumes that any files that are dragged and dropped into the main application window are LS-DYNA keyword files and will automatically try to open them.

Renumber

- **Case 25077**

It is now possible to renumber entities into any include ranges set for individual include files in the include renumber panel rather than having to renumber all include files.

- **Case 23114**

A new option has been added so that under normal operation PRIMER will avoid label clashes between entities which may lead to problems reading models into other pre-processers. For example, labels clashes between different element types.

- **Case 17771**

The include renumbering panel now provides additional information when there is a clash between label ranges of various include files

Rigid bodies

- **Case 24109**

An option has been added to the *CONSTRAINED_RIGID_BODY edit panel to swap the slave and master parts (PIDS and PIDM).

Scripting

- **Case 24682**

The position of the picking window can now be controlled from JavaScript by using Options.pick_window_position

- **Case 24651**

A File.Rename method has been added to enable files to be renamed.

- **Case 23747**

A new -js_arg command line argument has been added to PRIMER to enable arguments to be passed to a JavaScript.

- **Case 23508**

Model.Write() and Include.Write() now have an optional 'large' argument to allow large formal files to be written. There is also a new Model method UsesLargeLabels() that will return true if a model uses large labels.

- **Case 23504**

WidgetItems now have an index property which is the index of the WidgetItem in the 'parent' Widget (or undefined if not assigned to a Widget)

- **Case 23500**

Blank and Unblank methods have been added for the keyword classes (e.g. Shell and Part)

- **Case 23498**

An onBeforeShow event has been added to the Window class.

- **Case 23497**

A new static Window method MasterResolution() has been added to return the resolution of the master programme window.

- **Case 23420**

A new Tshell class has been added to support thick shells

- **Case 23179**

A new Belt class has been added to JavaScript to allow seatbelt definitions to be created and edited from JavaScript.

- **Case 23177**

Connections can now be created, modified and deleted for Mechanisms in JavaScript.

- **Case 23176**

A new WidgetItem onMouseOver event has been added.

- **Case 23078**

A new static method Widget.StringLength(string) has been added that will return the length of a string in Widget units.

- **Case 22775**

A (read only) window property has been added to the Widget class so it is possible to find which Window a Widget is defined in.

- **Case 22355**

It is now possible to get/set point data and get assembly info in the Dummy and Mechanism classes.

- **Case 22347**

An onAfterShow event has been added to the Window class.

- **Case 22345**

A new showClose property has been added to the Window class to control whether the close (X) button is shown on the top right of the window.

- **Case 22344**

A new Window.Delete() method has been added which frees any User Objects in windows. This should not normally required and is only required for scripts that (re)create lots of windows.

- **Case 22343**

A new Widget.Delete() method has been added.

- **Case 22342**

An active property has been added to the Window class to (un)restrict input in a window.

- **Case 22341**

Property keepOnTop has been added to the Window class to allow users to keep a window on the top of the stacking order.

- **Case 22225**

A listbox widget has been added to JavaScript.

- **Case 21998**

Classes have been added to JavaScript for pretensioners, retractrors, seatbelts, sensors and sliprings

- **Case 21642**

ItemAt() and TotalItems() methods have been added to the Widget class to give access to individual WidgetItems defined in the Widget.

- **Case 21596**

File.FindFiles() can now find directories as well as files.

- **Case 20274**

Added java script support for keyword *DEFINE_CONNECTION_PROPERTIES

- **Case 19830**

The list of available scripts is now shown as a tree and the scripts can be organised into branches

- **Case 18984**

Support for *DAMPING_GLOBAL has been added to JavaScript

- **Case 18905**

Added 'Ejection mitigation' Script, calculating target locations on window (Forward/Rear) based on FMVSS 226 specification and export as csv file, there is an option to build model from csv file.

- **Case 18756**

Include file comments can now be edited from JavaScript.

- **Case 18748**

Methods TableProperties, AllTableProperties and FlaggedTableProperties have been added to the Part class to allow the user to get all of the part table properties for parts.

- **Case 14189**

The list of available scripts is now shown as a tree and the scripts can be organised into branches

- **Case 13592**

Added an element SPH class into PRIMER JavaScript.

Seatsquash

- **Case 24539**

Added -2 and -1 as valid ELFORM types when creating *INITIAL_FOAM_REFERENCE_GEOMERTY through seatsquash.

Sections

- **Case 23587**

Added *SECTION_SHELL type 52 and 54 to the drop down list on the section shell edit panel.

Selection

- **Case 24986**

When selecting elements by screen area PRIMER has historically used the test 'is the element centre in the area' to determine whether or not the element is selected.

This remains the default, but an option to change this test to 'element will be selected if any node is in the screen area' has been added.

- **Case 24316**

PRIMER 11 could be distinctly sluggish about screen-picking when model size exceeded ~10 million nodes and elements, and testing with a model of ~50m elements revealed it to be unacceptably slow.

Picking in PRIMER 12 has been optimised to improve this, partly by parallelising more of the screen-picking process, but also by improving the underlying algorithms. The difference won't be noticeable for small models, but for these very large models it is now 10x or more faster and much more usable.

- **Case 22803**

Materials can now be filtered in object (selection) menus by the various _ADD options (erosion, etc)

3.2 D3PLOT

3.2.1 Enhancements in 12

Annotation

- **Case 14298**

Under Utitlities->Entity Names, names can now be defined for certain entity types and can be displayed on the model. This can be used to annotate the model.

These user defined names can be saved and reused in later D3PLOT sessions.

Cut Section

- **Case 25805**

In version 12 the cut-section menu has been redesigned to make it easier to access the different options available.

A new option has also been added that can be used to exclude entities if different types form the cut-section. If for example you have a block of reinforced concrete that is modelled with solid elements and beams it is now possible to exclude the beams form the cut-section so only the solids are removed.

Data Access

- **Case 23875**

D3PLOT now supports the output files from models containing *ELEMENT_DISCRETE_SPHERE's run in R7.1.

- **Case 22078**

The components 2D max shear stress and strain (S2MAX_2D_SHEAR_STRESS and E2MAX_2D_SHEAR_STRAIN) have been added.

Deform

- **Case 24017**

An option has been added to the DEFORM->REF NODE menu to output coordinates in the WRITE menu as [undeformed coordinate] + [displacement in local system].

Envelope

- **Case 21293**

The Contact menu did not have a button to create and Envelope plot. This has now been added in.

In previous versions, if you wanted to do an envelope plot of a contact component, you had to select the component under the Contact menu, then go back to the 2D/3D menu and select the Envelope button there.

General

- **Case 25309**

A new option has been added to write the parent directory and filename of the model to the graphics window header. This can be set in the top menu Display->Options.

Graphics

- **Case 25806**

In version 12 beam offsets are taken into account when beams are drawn using the 'True Section' option and a ZTF file written by PRIMER 12 containing the offset information is present.

- **Case 25361**

The number of decimal places shown for the time in the bottom right hand corner of the graphics window can now be controlled in Disp Opts->Window Dressing.

- **Case 23971**

The maximum number of characters displayed for the header can now be set in the 'Disp Opt->Window Dressing' menu.

Also, the following preferences have been added to control what is displayed in the graphics window:

```
d3plot*plot_header
d3plot*plot_contour_bar
d3plot*plot_clock
d3plot*plot_date
d3plot*plot_triad
d3plot*header_type
d3plot*header_nchars
```

- **Case 23878**

In version 12 the way that data values are displayed when using dynamic labelling has been changed. By default the values will be displayed using 'Automatic' formatting but if the user changes the format used to display the contour bar values then the same format will also now be used for the data values.

- **Case 22968**

An option has been added to turn off the display of the numbers on the graticule.

Measure

- **Case 19851**

Measurements can now be given a label which is displayed alongside the measurement in the graphics window.

Multiple Windows

- **Case 23761**

If models are open in multiple windows it is now possible to move them all into window 1. A new button has been added to the Window->Edit Window menu on the top menu to do this.

Nastran

- **Case 21865**

The list of loadcases written in the bottom left-hand corner of the graphics window from a Nastran analysis can now be turned on or off with the DISP OPT->NASTRAN LC LIST.

Preferences

- **Case 25631**

D3PLOT now has a '-pref' command line option that can be used to select a user defined preference file that is read after any other preference files.

```
-pref=filename
```

If 'filename' contains a PATH then it should be the complete path to the file, if it doesn't contain a PATH then it is assumed that the file is located in the installation directory.

- **Case 24983**

A preference has been added to set the cut section display type;

```
d3plot*cut_section_display.
```

It can be set to OFF (default), WIREFRAME or TRANSPARENT.

- **Case 24459**

A new preference

```
d3plot*open_models_in_w1
```

has been added to open models in Window 1 by default rather than the next window.

Read

- **Case 25505**

In version 12 of D3PLOT models can be opened on a PC by using drag and drop. D3PLOT assumes that any files that are dragged and dropped into the main application window are LS-DYNA output files and will automatically try to open them.

- **Case 25494**

Recent files that have been read within D3PLOT are now available via a dropdown on the file read input text box.

- **Case 24459**

A new preference

```
d3plot*open_models_in_w1
```

has been added to open models in Window 1 by default rather than the next window.

Scripting

- **Case 24362**

A GetCutCoords() javascript function has been added to get the coordinates where the cut-section cuts through an element.

- **Case 19649**

The GetNumberOf() function can now be called with the argument GROUP to get the number of groups in a model.

A new function GetGroupInfo() has been added to get information about a group returning an object with properties .label and .name.

These can be used in conjunction with the /GROUP dialogue commands to manipulate groups in a script.

- **Case 18492**

The javascript function GetElemsInPart() has been added to get the elements in a part.

User Interface

- **Case 23773**

In version 12 the maximum length for filenames (including the directory) has been increased from 256 to 512 characters

NOTE : Internally Windows supports filenames of up to 32768 but currently it does not support pathnames longer than 260 characters within Windows Explorer or the command line interface.

X Sections

- **Case 20832**

If X-sections have been named using the *DATABASE_CROSS_SECTION_ID option the names are now displayed alongside the cross section ID in the blank menu.

XY Plotting

- **Case 23803**

A new XY Data plotting button has been added to the quick pick menu when right-clicking (or dragging) on an entity.

An XY plot of the data component selected in the data menu is plotted for the selected entities.

ZTF File

- **Case 22116**

D3PLOT will now issue a once-per-session warning if MAXINT is not set to the default of 3 and the user is trying to plot by TOP, MIDDLE or BOTTOM surface and there is no ZTF file as in this situation D3PLOT may not be able to determine which is the correct integration point to read without the additional data in the ZTF file.

3.3 T/HIS

3.3.1 Enhancements in 12

Annotate

- **Cases 17684, 20270 22156 and 23865**

Annotations can now be created on graphs (Measure->Annotate). They can be made on curve points or screen points.

Curves

- **Case 19108**

The curve 'History' option in Quick Pick now contains the option to change the model that data is being read from. Previously you could only change the entity ID and component before updating the curve.

FAST-TCF

- **Case 24591**

In previous versions T/HIS would output a command to label every curve generated by a FAST-TCF script and for most curves the command would usually contain the default curve label generated when T/HIS read in the data.

If when running a FAST-TCF the wrong data was somehow read in the user might fail to notice this as the default curve label that would show the incorrect data source would be overwritten with that written to the FAST-TCF script when it was generated.

From version 12 onwards T/HIS now keeps a track of the curve labels and when it writes a FAST-TCF script it checks to see if the curve label has been modified interactively by the user. If it has then the command to label the curve is written out as normal. If the label is still the default one generated by T/HIS then it is now output as a comment line only.

```
 #
 # Curve labels - (modified by user)
 #
 # label curve_12 K.E. - Part PART1002 : (knee Bolster springs)
 # label curve_13 K.E. - Part AAAAAAAA : (STEERING WHEEL RIM BEAMS @ STEERING
WHEEL RIM)
 # label curve_14 K.E. - Part 1100 : (arup bulk head beam)
 label curve_15 XXXXX K.E. - Part 1300 : (rigid top of steerin g column)
 # label curve_16 K.E. - Part 1302 : (STEERING WH RIM SHELLS @ STEERING WH
RIM SHELLS)
 label curve_17 XXXXX Disp x - Node 100437 : (Sled)
 # label curve_18 Disp x - Node 200000 : (Head)
 # label curve_19 Disp x - Node 100437 : (Sled)
 # label curve_20 Disp x - Node 200000 : (Head)
 # label curve_21 Disp x - Node 201786 : (Chest)
 # label curve_22 Disp x - Node 203303 : (Pelvis)
```

- **Case 22271**

T/HIS 12.0 has a new JavaScript function that can be used to get the value of a FAST-TCF variable.

```
var value = GetFtcfVar('job')
```

would return the value of the FAST-TCF variable 'job'.

(Note: all FAST-TCF variables are case insensitive).

- **Case 22259**

T/HIS 12.0 has a new JavaScript function that can be used to set the value of a FAST-TCF variable.

```
var value = SetFtcfVar('run_number','20')
```

would set the value of the FAST-TCF variable 'run_number' to '20'. If the variable already existed then it's value would be updated otherwise it would be created.

(Note: all FAST-TCF variables are case insensitive).

Graphics

- **Case 21065**

If a large number of curves were selected for a curve operation by dragging across them in a graph T/HIS could take some time to highlight all the curves in the graph and in the curve menu.

In version 12 this process has been speeded up.

- **Case 4246**

The Display menu now contains options to turn on and off the drawing of a graphs top and right hand axes. By default these are both set to ON.

As well as turning on and off the axes via the Display menu there now 2 new preference settings

```
this*axis_top: ON / OFF
this*axis_right: ON / OFF
```

Measure

- **Case 20672**

A new menu has been added to the tools panel for making measurements on graphs (shortcut key 'm').

- Multiple measurements can be displayed at the same time
- Measurements can be made between curve points and/or screen points
- Single points can be measured too
- The measured values can be displayed on the graph
- Labels can be added to the measurements
- There is an annotation mode for annotating your curves
- The formatting of the measurements are user configurable

Operations

- **Case 24032**

The Weighted Integrated Factor method has been added to calculate the correlation between two curves (menu Automotive->WIF).

- **Case 23254**

The Rolling Average function (R-AVE)has been modified so that it now has a user defined time window that it averages across.

By default the time window is set to 0.0 which means 'infinite' so the default output from the function is unchanged. If the user sets the window time to T, then for each curve point the y value is calculated by averaging between -T/2 and T/2.

- **Case 22899**

There are now two different scaling options that can be applied when doing an FFT.

Scaling option 1 - this is the default and recommended for most purposes. It is consistent with other signal processing software, giving a magnitude independent of any padding required to make the input curve have a number of points equal to a power of 2. Performing an inverse FFT on the resulting curves will NOT get back exactly to the original curve if it did not have a number of points equal to a power of 2.

Scaling option 2 - with this option, applying an inverse FFT to the resulting curves will generate a curve the same as the original even if it did not have a number of points equal to a power of 2. This is useful if users wish to create their own filters, where the filter characteristic is defined in the frequency domain.

- **Case 15904**

In version 12 the CLIP function has been modified so that when a curve is clipped new points are now inserted at the clip value by interpolating between the existing curve points. Previously any points with x-axis values outside the clipping region were thrown away and any points with y-axis values outside the region had the y-axis values adjusted to the clip region boundaries.

- **Case 5417**

The CLIP function now has options to pick the x and y min/max values interactively by clicking on the screen. Each value can be picked individually or an area defining the limits can be dragged out.

When picking points there is also an option to select the nearest curve point to where you click instead the actual screen coordinate.

Preferences

- **Case 25632**

T/HIS now has a '-pref' command line option that can be used to select a user defined preference file that is read after any other preference files.

```
-pref=filename
```

If 'filename' contains a PATH then it should be the complete path to the file, if it doesn't contain a PATH then it is assumed that the file is located in the installation directory.

Properties

- **Case 25061**

In version 11 it was possible to change the colour used to display injury criteria values (HIC, PHD etc) but it wasn't obvious how to do this (it is controlled by the Properties->Format menu).

In version 12 there are now 'Properties' buttons next to the options where the display of injury criteria are turned on and off which are a shortcut to the Format menu to make it easy to access these settings.

The option to control the colour has also been split up so there are now 2 separate settings:

'Text Colour' controls the colour of the injury criteria values and labels.
'Injury Marker Colour' controls the colour of the lines drawn for each injury value.

By default these are both set to the current foreground colour.

2 new preferences have also been added that can be used to set the defaults for these 2 options.

```
this*injury_line_colour:
this*injury_text_colour:
```

Quick Pick

- **Case 22255**

The Quick Pick menu in version 12 of T/HIS has 2 new options, 'Copy Curve(s)' and 'Paste Curves'. The COPY option can be used to select curves. Once curves have been selected with COPY the PASTE option can then be used to make a copy of the curves and to unblank them in just the graph that PASTE was selected in.

If for example you used COPY in graph 1 and dragged across curves 4 and 5 and then selected PASTE in graph 2 then 2 new curves would be created which were exact copies of 4 and 5 and would be unblanked in just graph 2.

- **Case 19019**

The Quick-Pick Operate functions popup now contains the Clip function. The CLIP function entry also has a new popup submenu that displays the x and y clipping values and can be used to change those values.

Read

- **Case 25506**

In version 12 of T/HIS models can be opened on a PC by using drag and drop. If files are dragged and dropped into the main application window T/HIS will check if the file is one of the LS-DYNA output files it reads and if it is then it will open all of the model files that it can find that go with the file. T/HIS also assumes that any files with a .cur file extension are T/HIS curve files and tries to read them. Any files with a .csv file extension are treated as CSV files and the CSV read menu is automatically displayed.

- **Case 25496**

Recent files that have been read within T/HIS are now available via a dropdown on the file read input text box.

- **Case 25462**

Version 12 of T/HIS now has separate data components for reading the master and slave side contact forces. Previously T/HIS just has buttons for X,Y,Z and Magnitude and would always report the master side forces for contacts. Previously the master side was selected as in old versions of LS-DYNA node to surface contacts only reported forces on the master side. Recent versions of LS-DYNA can generate forces on the slave side of some contacts (FORCE_TRANSDUCERS) while the master side reports zero force. Because of this T/HIS now has separate options for master and slave forces.

- **Case 25009**

The CSV file reader in T/HIS has been modified so that the field separator can now be changed from the default comma to either a tab or spaces.

If the 'space' option is used then multiple spaces are counted as a single field separator. If curve or axis labels are defined in the file and they contain spaces then they need to be enclosed in pairs of " quotes.

A new FAST-TCF command has been added to set the separator used when reading in a CSV file

```
define csv_separator tab
```
(options are comma, space or tab)

A new preference has also been added that can be used to change the default separator from 'comma' to either 'space' or 'tab'

```
this*csv_separator: Tab
```

- **Case 24634**

T/HIS 12.0 supports the 'Eroded Hourglass Energy' data component that is written to the GLSTAT ASCII file and LSDA (binout) file by some versions of LS-DYNA

- **Case 24523**

T/HIS 12.0 now supports the following INTOUT and NODOUT options on *DATABASE_BINARY_EXTENT

INTOUT : STRESS

> This option generates stress tensor results for all integration points. For fully integrated shells this will generate results for all 4 in-plane integration points at each through thickness point. For fully integrated solids data for 8 integration points will be generated.

NODOUT : STRESS

> This option will output stress tensors at the nodal points. These values are generated by extrapolating results from the integration points.

- **Case 23338**

T/HIS now supports the NODOUT option on *DATABASE_EXTENT_BINARY for writing element data values that have been extrapolated from the integration points to the nodes. This option applied to Solids, Shells and Thick Shells.

Currently only the STRESS, STRAIN and ALL options are supported.

- **Case 22139**

In Version 11 if you had associated '.cur' files with T/HIS then double clicking on a '.cur' file on a PC would start T/HIS and read the curve in. If you then wanted to read in another curve file you had to change the READ menu to CURVE before you could select the 2nd curve file.

From version 12 onwards the READ menu will automatically select the CURVE option if a curve file is read in when T/HIS starts by double clicking on the curve file.

- **Case 20557**

In some recent versions of LS-DYNA the shell and thickshell results in the LSDA (binout) file can be written using the Global coordinate system instead of the default local coordinate system (see EOCS on *CONTROL_OUTPUT). If the LSDA file also contains information for fully integrated shells and thickshells with multiple in-plane integration points then the file contains both ELOUT and ELOUTDET but the ELOUTDET results ignore the EOCS option and are still written using the local coordinate system.

If T/HIS detects both ELOUT and ELOUTDET in the LSDA file and they are using different coordinate systems T/HIS now displays a warning when the model is opened. By default when reading data T/HIS uses the ELOUTDET file if both are present as normally the data is ELOUT is a subset of that in ELOUTDET. If T/HIS detects a mix of coordinate systems it now offers the user a choice to ignore the ELOUTDET file when reading in Shell and Thickshell data so that the Global values in ELOUT can be read.

- **Case 12809**

If a user accidentally clicked on the 'Read > Keyboard' option then it wasn't obvious how to get out of the window that appears into which the curve information is entered.

Version 12 now has a 'Quit' button which closes the window without storing any curve data that has been added and a 'Finish' option that becomes active as soon as the 1st data point has been entered and saves the curve to the 1st free curve.

- **Case 11012**

If LS-DYNA incorrectly split the output for a single state across multiple THF files then T/HIS could fail to read the files correctly.

Scripting

- **Cases 23511**

A Datum class has been added to the T/His JavaScript API.

User Interface

- **Cases 18077 and 23772**

In version 12 the maximum length for filenames (including the directory) has been increased from 256 to 512 characters

NOTE : Internally Windows supports filenames of up to 32768 but currently it does not support pathnames longer than 260 characters within Windows Explorer or the command line interface.

3.4 REPORTER

3.4.1 Enhancements in 12

Capture

- **Case 21309**

REPORTER will now prompt the user to replace variables in the macro after doing a capture in PRIMER. For each matched text string you can choose whether to replace it with a variable or you can do 'Yes to All' or 'No to All'

Scripting

- **Case 23990**

A new Batch() method has been added to JavaScript so scripts can test if REPORTER is running in batch mode

- **Case 22974**

New Item and Page classes have been added to JavaScript.

- **Case 17602**

A Duplicate method has been added to the Page class in JavaScript

- **Case 15772**

A System command has been added to JavaScript

User interface

- **Case 23503**

Objects can now be locked to prevent them from being moved on the page