Contents

Overview of Visualise 3D ................................................................. 3
About Theorem .............................................................................. 3
Theorem’s Product Suite ................................................................. 3
The Visualize 3D CATIA 3DEXPERIENCE to JT Bi-directional Translator .................. 4
Primary Product Features .................................................................. 4
Primary Product benefits? ............................................................... 5
Getting Started .................................................................................. 6
Documentation ................................................................................. 6
Installation Media ............................................................................. 6
Installation & License Configuration .................................................. 6
Running the Product .......................................................................... 7
Using the Product ............................................................................... 8
Translating Interactively from within 3DEXPERIENCE ................................. 8
Theorem Interactive Conversion Settings ........................................... 8
Import JT .......................................................................................... 13
Export JT .......................................................................................... 14
Property Mapping Files .................................................................... 15
PMI Type Filter Files ......................................................................... 16
Advanced User Options ..................................................................... 17
Theorem Export ............................................................................... 19
JT Template files .............................................................................. 21
Theorem Import ............................................................................... 23
Translating in Batch using CATUtil – PLMBatchDataExchange ....................... 26
Batch Export to JT ........................................................................... 26
Batch Import from JT ....................................................................... 27
Log File Generation ......................................................................... 28
Export process Log Files .................................................................. 28
Import process Log Files ................................................................... 29
CATIA 3DEXPERIENCE Environment Files ............................................. 29
Appendix A – JT Configuration File .................................................... 30
Introduction........................................................................................ 30
The Setup Section .................................................................................................................. 30
The Level of Detail Section ..................................................................................................... 33
The Filter Section ................................................................................................................... 33
The Metadata section ............................................................................................................ 34
Appendix B – Large Assembly Processing (LAP) ................................................................. 35
Overview ................................................................................................................................ 35
LAP Options ............................................................................................................................. 35
Interactive Operation .............................................................................................................. 36
DataExchangePLMBatch Operation ....................................................................................... 36
Batch Mode Operation ............................................................................................................ 36
Master Template ...................................................................................................................... 37
Appendix C – Batch Processing ............................................................................................. 38
Appendix D – Creating a Login Ticket ................................................................................... 40
Overview of Visualise 3D

About Theorem

Theorem Solutions is a world leader in the field of Engineering Data Services and Solutions. This leadership position stems from the quality of our technology and the people in the company. Quality comes not only from the skills and commitment of our staff, but also from the vigorous industrial use of our technology & services by world leading customers.

We are proud that the vast majority of the world's leading Automotive, Aerospace, Defense, Power Generation and Transportation companies and their Supply chains use our products and services daily. Working closely with our customers, to both fully understand their requirements and feed their input into our development processes has significantly contributed to our technology and industry knowledge.

Theorem Solutions is an independent UK headquartered company incorporated in 1990, with sales and support offices in the UK and USA. Theorem has strong relationships with the major CAD and PLM vendors, including; Autodesk, Dassault Systemes, ICEM Technologies (a Dassault company), PTC, SolidWorks, Spatial Technology and Siemens PLM Software. These relationships enable us to deliver best in class services and solutions to engineering companies worldwide.

Theorem’s Product Suite

The Visualize 3D CATIA 3DEXPERIENCE to JT translator is part of our Visualize 3D brand which is one of 5 core Theorem brands which consist of:

- **CADverter**
  Direct translation of 3D data to or from an alternate CAD, Visualization or Standards Based format

- **Multi-CAD**
  Interactive integration of non-native 3D data formats into the native CAD system

- **Visualize 3D**
  Direct translation of 3D data for the purpose of Visualization
The Visualize 3D CATIA 3DEXPERIENCE to JT Bi-directional Translator

The CATIA 3DEXPERIENCE to JT translator may be installed on a number of machines each accessing a central network-floating license.

The CATIA 3DEXPERIENCE to JT Translator is a bi-directional direct database converter between the Dassault Systemes CATIA 3DEXPERIENCE Modelling Application and the JT file format, used by the Siemens Teamcenter Visualization products.

It enables the user to convert all forms of 3D Mechanical Design Geometry and Assembly data, together with system defined attribute information and colour information, between these two systems. This product is designed for companies using 3DEXPERIENCE who have selected JT to be their main method of collaboration and communication between OEMs and their customers or suppliers.

It is also a major method of visualization and is used by companies using JT based solutions to translate their CATIA 3DEXPERIENCE data into the JT format.

The translator can be invoked in either an interactive or batch mode.

Primary Product Features

- Converts all types of geometry, wire frame, surfaces, trimmed surfaces (faces) and solid models.
- Converts assembly structure between both systems.
- Converts attribute data including colour and layer information.
- Integrated with the CATIA 3DEXPERIENCE installation.
- The conversion process can be run Interactively or in Batch Mode
- Data can be filtered by layer and entity type during processing. Geometry can be filtered and selectively processed.
• Uses the CATIA 3DEXPERIENCE API and Siemens JTOpen API to read and write data.

• In creating JT files a number of data types can be generated. A facetted representation, a JTBrep definition or an XTBrep definition. As standard a facetted representation is created with an option to select whether JTBrep or XTBrep definition is created.

Primary Product benefits?

• Being a direct database converter all pre and post processing is eliminated, saving time.

• Reduce costs due to processing time and increase overall conversion success levels by filtering input data and focusing the conversion to only those elements required.

• Reduce costs and risks associated to accessing the wrong version of data by integrating the conversion process into a related business processes.

• With over 20 years of industrial use Theorem translation products robustness and quality is well proven, reducing your business risk.

This document will focus specifically on guidance for the use of the Visualize 3D CATIA 3DEXPERIENCE to JT product. For information regarding any of Theorem’s product ranges please contact sales@theorem.com
Getting Started

Documentation
The latest copy of this documentation can be found on our web site at:

http://www.theorem.com/Documentation

Installation Media
The latest copy of Theorem software can be found via our web site at:

http://www.theorem.com/Product-Release-Notes

Each product has a specific link to the Product Release Document, which contains a link to the download location of the installation CD.

Alternatively, you can request a copy of the software to be shipped on a physical CD.

Installation & License Configuration
The installation and license configuration of this product are run from separate .msi packages. More information regarding these steps can be found in the following document

CATIA V6 to JT Installation and licensing Documentation
Running the Product
Once configured and licensed, the product is ready to be run.

There are 3 distinct ways of running the translator:

- **Interactively from within CATIA 3DEXPERIENCE**
  - The Interactive Interface provides a direct method of importing and exporting to and from 3DEXPERIENCE.

- **In Batch via CATUtil - DataExchangePLMBatch**
  - The 3DEXPERIENCE DataExchangePLMBatch Interface provides a direct method of invoking the translator. It can be used on an **On Demand** basis to translate single or multiple files.

- **On the command line**
  - For **export only** at this stage a command line method of invoking the translator is possible, further information regarding this can be found in the appendices B, C & D which also cover Large Assembly Processing and the creation of job files for the translator.
Using the Product

Translating Interactively from within 3DEXPERIENCE

The CATIA 3DEXPERIENCE to JT translator allows an opened CATIA 3DEXPERIENCE part or Assembly to be exported directly to JT, and for a JT part or assembly to be imported directly into the CATIA 3DEXPERIENCE application.

In order to translate from within CATIA 3DEXPERIENCE, the CATIA 3DEXPERIENCE application must be started using a Theorem environment, so that the appropriate Theorem partner plug-ins are available. (See CATIA 3DEXPERIENCE Environment files)

CATIA 3DEXPERIENCE can be started from a desktop shortcut created at installation time. E.g.

Alternatively, it can be started via the script provided in the Translator installation located in:

\langleinstallation_directory\rangle\bin

The script name is:

\textit{start\_3DEXPERIENCE\_Theorem\_Multi\_CAD\_JT\_CATIAV6R\langleversion\rangle\.cmd}

(where \langleversion\rangle should be substituted for the version of CATIA 3DEXPERIENCE that you have installed – e.g. 2013x, 2015x, 2016x, 2017x):

Theorem Interactive Conversion Settings

The 3DEXPERIENCE interface offers a number of settings to enable configuration of the Translation Process.

The method for altering these settings differs between 3DEXPERIENCE versions.

For R2013x, the User settings are accessed through the \textbf{Tools}->\textbf{Options} menu:-
for R2015x and later revisions these are accessed through the User Preferences options:

Of importance for all versions is the Compatibility options page as shown here for R2013x:
And here for R2105x and R2016x, R2017x :-

This page sets the preferred mode of conversion (in this case Exact), and also the general options “Convert coordinate systems” and “3D Annotation”.

Two dedicated pages of options under the ‘Tools>Options>Infrastructure>Theorem>MultiCAD JT’ tabs allow the user import and export operations is available under the ‘Infrastructure’ settings and is presented as follows:-

For R2013x:-

The Export JT Tab
...and the Import JT tab

For R2015x, R2016x, and R2017x:-

The Export Tab
...and the Import JT tab

Each of the options in these panels is described below:
## Import JT

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conversion Elements - Points</strong></td>
<td>Enables Point entities to be retained. Default is OFF.</td>
</tr>
<tr>
<td></td>
<td>Enabling this option will read all Point entities Command Line Syntax:</td>
</tr>
<tr>
<td></td>
<td>o Option data file jt_xcad_opts.txt syntax:</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>read_points</strong> – to turn on</td>
</tr>
<tr>
<td><strong>Conversion Elements – Wire Frame</strong></td>
<td>Process Wireframe entities during translation (if present).</td>
</tr>
<tr>
<td></td>
<td>Default is OFF</td>
</tr>
<tr>
<td></td>
<td>o Option data file jt_xcad_opts.txt syntax:</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>read_wire_frame</strong></td>
</tr>
<tr>
<td><strong>Conversion Elements – Structure</strong></td>
<td>Enables Assembly Structure to be retained. Default is ON.</td>
</tr>
<tr>
<td></td>
<td>Disabling this option will remove all assembly structure and collapse ALL geometry into a single selectable object</td>
</tr>
<tr>
<td></td>
<td>o Option data file jt_xcad_opts.txt syntax:</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>no_structure</strong></td>
</tr>
<tr>
<td><strong>Property Mapping File</strong></td>
<td>Filter name choices can be set via a configuration file:</td>
</tr>
<tr>
<td></td>
<td>%TS_INST%data\jt\jt_mcad_options_configuration.txt</td>
</tr>
<tr>
<td></td>
<td>The default mapping is located here:</td>
</tr>
<tr>
<td></td>
<td>%TS_INST%data\jt\jt_v6_property_mapping.txt</td>
</tr>
<tr>
<td></td>
<td>This is a file containing a list of CAD properties and information on how they are mapped from the JT file. (See section Property Mapping Files)</td>
</tr>
<tr>
<td></td>
<td>The associated “View” button will present the selected filter file content to the user in an editor window.</td>
</tr>
<tr>
<td><strong>PMI Filter File Name</strong></td>
<td>Filter name choices which are set via a configuration file:</td>
</tr>
<tr>
<td></td>
<td>%TS_INST%data\jt\jt_mcad_options_configuration.txt</td>
</tr>
<tr>
<td></td>
<td>The default mapping is located here:</td>
</tr>
<tr>
<td></td>
<td>%TS_INST%data\jt\jt_pmi_filter.txt</td>
</tr>
<tr>
<td></td>
<td>This file lists JT PMI type names which are to be ignored by the import process. (See section PMI Type Filter Files)</td>
</tr>
<tr>
<td></td>
<td>The associated “View” button will present the selected filter file content to the user in an editor window.</td>
</tr>
</tbody>
</table>
### Export JT

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Precise BREP Format</strong></td>
<td>This option allows the user to select the format of precise data that is written to the JT file</td>
</tr>
<tr>
<td></td>
<td>- XT-BREP <em>(Default)</em></td>
</tr>
<tr>
<td></td>
<td>- Creates XT-BREP (Parasolid type embedded solid body definitions) in the JT output files.</td>
</tr>
<tr>
<td></td>
<td>- JT-BREP</td>
</tr>
<tr>
<td></td>
<td>- Creates Siemens JtBrep data</td>
</tr>
<tr>
<td><strong>Conversion Elements - Bodies</strong></td>
<td>Enables CATIA Bodies to be processed <em>(Default is ON)</em></td>
</tr>
<tr>
<td><strong>Conversion Elements - Surfaces</strong></td>
<td>Enables CATIA independent Surfaces to be processed <em>(Default is ON)</em></td>
</tr>
<tr>
<td><strong>Conversion Elements - Curves</strong></td>
<td>Enables CATIA Wireframe Curves to be processed <em>(Default is ON)</em></td>
</tr>
<tr>
<td><strong>Conversion Elements - Axes</strong></td>
<td>Enables CATIA Axes systems to be processed <em>(Default is ON)</em></td>
</tr>
<tr>
<td><strong>Conversion Elements - Points</strong></td>
<td>Enables CATIA Points to be processed <em>(Default is ON)</em></td>
</tr>
<tr>
<td><strong>Conversion Elements - PMI</strong></td>
<td>Enables CATIA PMI/FTA data to be processed <em>(Default is ON)</em></td>
</tr>
<tr>
<td><strong>Conversion Elements – No Show</strong></td>
<td>Enables CATIA entities in No show (Hide) state to be converted. <em>(Default is OFF - i.e. to ignore hidden entities)</em></td>
</tr>
<tr>
<td><strong>Sub-Part Data</strong></td>
<td>The user can control how Sub Part Data is processed. There are 3 options:</td>
</tr>
<tr>
<td></td>
<td>- None - Multiple bodies in a single CATIA part representation will result in a single leaf node per CATIA Part in the JT representation. <em>(Default Option)</em></td>
</tr>
<tr>
<td></td>
<td>- Body Names - The “Body Names” option adds a level of sub structure that can be accessed using the Show Leaf Structure in Jt VisMockup. It creates a leaf node for each CATIA body that is converted.</td>
</tr>
<tr>
<td><strong>JT config File</strong></td>
<td>JT configuration file name choices and selection are set via a configuration file:</td>
</tr>
</tbody>
</table>
The default configuration file is located here:

```
%TS_INST%\data\jt\jt_mcad_options_configuration.txt
```

Please see [Appendix A](#) for a full description of the JT config file. The associated “View” button will present the selected configuration file content to the user in an editor window.

### Property Mapping File

Filter name choices which are set via a configuration file:

```
%TS_INST%\data\jt\jt_mcad_options_configuration.txt
```

The default configuration file is located here:

```
%TS_INST%\data\jt\jt_v6_jt_property_mapping.txt
```

This is a file containing a list of CAD properties and information on how they are mapped to the JT file. (See section [Property Mapping Files](#)) The associated “View” button will present the selected filter file content to the user in an editor window.

---

**Note!** Administrator control locking is available for this tools option page and is controlled by specification of a `CATReferenceSettingPath` and running CATIA with the `–admin` option. Please refer to standard CATIA 3DEXPERIENCE documentation for more information.

**Note!** It is important that the selections on this option page are saved in the user’s CATSettings data prior to executing conversions by clicking the “OK” button at the bottom of the page.

### Property Mapping Files

The selections of the displayed JT Config File (.config) Property Mapping files and PMI Type Filter Files are set through the configuration file:

```
%TS_INST%\data\jt\jt_mcad_options_configuration.txt
```

The format of the `jt_mcad_options_configuration.txt` is:

```
<jt_config_files>
Default TessCATIA6MultiCAD;%TS_INST%\etc\tessCATIAV6MultiCAD.config
</jt_config_files>

<jt_import_property_files>
Default Import Jt Property Filter;%TS_INST%\data\jt\jt_v6_property_mapping.txt
</jt_import_property_files>

<jt_export_property_files>
Default Export Jt Property Filter;%TS_INST%\data\jt\v6_jt_property_mapping.txt
</jt_export_property_files>

<jt_import_pmi_files>
Default Import PMI Type Filter;%TS_INST%\data\jt\jt_pmi_filter.txt
</jt_import_pmi_files>
```

---

©Theorem Solutions 2017
There is one option menu entry per line containing <Description> ; <Absolute File Path>

Where the <Description> is the text to be displayed in the option menu and the file path is the location of the JT write configuration file or the property filter. This path definition can include environment variables.

The user can control the mapping of user defined attributes contained in the PLM part definition and external files during the import and export processes.

The ‘JT Import Property Mapping File’ and ‘JT Export Property Mapping File’ are text files of a format described below:

A mapping file is used to control which properties are converted by setting a control value. Setting the control value to 0 will stop a specific property from being exported.

The mapping file can also enable the mapping of property names to new names: this is performed by switching the name between the input name (= field 1) and the output name (= field 2)

The File Line Format is as follows:

SourceName, TargetName, Control, Dummy, Dummy, Dummy
Lines beginning with a ‘#’ are taken as comment lines
SourceName – is the input attribute name.
TargetName – is the output attribute name (NULL means use SourceName)
Control – is flag to control conversion: 0 - Do not convert, 1 – Do convert
Dummy – unused fields
If SourceName is given as NULL then any item not included in map will match
So to include all other attributes use
NULL,NULL,1,,,
Or to exclude all other attributes use
NULL,NULL,0,,
Examples
To exclude the MPARTNAME attribute
MPARTNAME,NULL,0,,
To include the TAG attribute
TAG,NULL,1,,
To rename the REFASSYTYPE attribute to ASSYTYPE
REFASSYTYPE,ASSYTYPE,1,,

PMI Type Filter Files

The user can control the filtering of PMI types on import from JT by specifying an appropriate filter file. A default filter file is provided with the installation located as %TS_INST%\data\jt\jt_pmi_filter.txt. This file contains a list of all PMI types by name, and can be edited to exclude different named types by removing a ‘#’ (comment character) from the type name not required to be imported.
e.g. if the file is edited as follows

```
# File for filtering on PMI type via the command : pmi_filter_file "file name"
#
# Line Format:-
# "pmi type"
#
# Lines that start with a '#' are ignored.
# The supplied file contains all possible PMI types in alphabetical order preceded by a '#'
#
# To prevent a particular PMI type from being processed, remove the '#' from that type
#
#arc spot weld
#attribute note
#balloon
#bead
#bundle dressing note
#callout dimension
#center point
#centerline
#cert point
#chamfer

. . .
```

Any PMI entities of type ‘center point’ would NOT be imported.

This facility was introduced to reduce processing time due to large numbers of ‘redundant’ PMI entities in a JT file.

Note that default settings that can control the JT Export plugin, are also read from the data file `%TS_INST%\data\jt\xcad_jt_opts.txt`.

Default settings that can control the JT Import plugin, are also read from the data file `%TS_INST%\data\jt\jt_xcad_opts.txt`.

**Advanced User Options**

An advanced option panel is available by setting the environment variable in the relevant environment file

e.g. `C:\Program Files\Theorem\20.2\B214\win_b64\CATEnvTheorem_Multi-CAD_JT_CATIAV6R2013x.txt`

```
TS_JT_MCAD_OPTIONS_ADVANCED_USER=1
```

Currently this setting is only applicable for the Export to JT settings.
Here for example is the R2017x presented options:

![Preferences window](image)

Additional options are displayed that allow the specification of the JT File Structure created:

Options are:

- As Configured, MONOLITHIC, PER_PART, FULL_SHATTER or MIMIC

An option is provided to override the configured JT LOD settings.

If the Number of LODS is set to a value of 0, the LOD settings contained in the selected JT Config File will be used.

Please see Appendix A for a full description of the JT Config file.
Visualize 3D v20.2 for CATIA V6 - JT

Theorem Export

Once CATIA 3DEXPERIENCE has been started it is possible for an opened product to be exported to JT.

If CATIA 3DEXPERIENCE R2013x is being used, in order to export, the user selects the PLMAccess > Export option as shown here:-

If CATIA 3DEXPERIENCE R2015x or later is being used, the user selects the menu item as shown here:-
This menu action will then present the user with a Save As dialog box similar to this:

For R2013x:
For R2015x:

![Image of Export settings]

In all cases the “Save as types:” or “Format” selection should be set to JT_THEOREM (*.jt).

The Save or OK button can now be pressed to initiate the conversion to the selected output file name and location.

JT Template files

If, when a representation is read from CATIA 3DEXPERIENCE, no geometry is found in the representation, all the geometry is hidden or a major write error occurs preventing a JT file being produced, then an empty JT file (named by default template_empty.jt, template_hidden and template.jt respectively) will be copied to the expected output file name.

This enables the conversion process to complete successfully and maintains the expected file outputs.
This behaviour is implemented via environment variables set in the Theorem CATEnv file e.g. %TS_INST%\B214\win_b64\CATEnv\Theorem_Multi-CAD_JT_CATIAV6R<ver>.txt file, typically as follows:

\texttt{TS\_JT\_TEMPLATE\_FILE=\textbackslash C:\Program Files\Theorem\20.2\data\jt\template.jt}

\texttt{TS\_JT\_TEMPLATE\_FILE\_EMPTY=\textbackslash C:\Program Files\Theorem\20.2\data\jt\template\_empty.jt}

\texttt{TS\_JT\_TEMPLATE\_FILE\_HIDDEN=\textbackslash C:\Program Files\Theorem\20.2\data\jt\template\_hidden.jt}

The user can change the content of these JT files or their location and names as required.
Theorem Import

Once CATIA 3DEXPERIENCE has been started it is possible for a model to be imported from JT.

In order to import, the user selects the following Menu Option:

For CATIA 3DEXPERIENCE R2013x:-

For 2015x and later:-
This launches the Theorem Import Menu:

For R2013x:-

For R2015x:-

For R2016x or later:-
On selecting *Open* or *OK* the on the File selection Panel the JT data will be translated and imported into the CATIA 3DEXPERIENCE database. It will also be opened in the user’s session.
Translating in Batch using CATUtil – PLMBatchDataExchange

The Dassault Systemes CATUTIL Batch Monitor tool can be launched with the correct Theorem enabled environment using the scripts provided in the Translator installation at:

```
<installation_directory>\bin\start_CATUTIL_Theorem_Multi-CAD_JT_CATIAV6R<revision>.cmd
```

(where [revision] should be substituted for the version of 3DEXPERIENCE that you have installed – e.g. 2013x, 2015x, 2016x, 2017x)

Note: It can also be started from Tools>Utility within an Interactive CATIA session.

The settings made in the user options page “MultiCAD JT” described in the interactive usage above, are also used in BATCH mode.

In some cases, it may be desirable to ignore the interactive settings while running in batch mode.

This can be achieved by setting the following variable in the CATIA Environment being used:-

```
TS_IGNORE_JT_CATSETTINGS=1
```

Batch Export to JT

Here is an example showing the appropriate settings in preparation for a database selection and an export “save” or “run”:-

Note selection of “Export” operation, usage “JT_THEOREM” and extension “jt”.
The Representation or Product for export can be found using the standard Enovia V6 selection tools similar to that shown below.

Batch Import from JT
Here is an example showing the appropriate settings in preparation for an input JT file selection and an export “save” or “run”:
Note selection of “Import” operation, usage “JT_THEOREM” and extension “.jt”.

The JT file for import can be found using the standard Enovia V6 selection tools similar to that shown below.

![Image of Enovia selection tools]

Log File Generation

Export process Log Files
In the process of exporting the selected CATIA 3DEXPERIENCE part or assembly, the following log files are generated by the JT Export plug-in.

The process log and error messages are recorded in a ‘.err’ file located in the CATIA 3DEXPERIENCE CATReport directory. The file is named after the active CATIA component.

  e.g. %CATReport%\model1.err

Additional log files are created in the TSC_TEMP_DIR directory. This directory is defined in the %TS_INST%\ts_env.bat file.

  %TSC_TEMP_DIR%\Read_to_viewer_<input_part_name>.log

Where: <input_part_name> is the name of the input part (or the active part name in interactive usage)
This contains information describing the CATIA 3DEXPERIENCE 'data read' processing into Theorem Intermediate data format

  %TSC_TEMP_DIR%\viewer_<part-name>_screen_output.log

Where <part-name> is the selected output file name
This contains the screen output of the process of writing the data to JT.

  %TSC_TEMP_DIR%\viewer_<part-name>.log

Where <part-name> is the selected output file name
This contains detailed process information of the write of the data to JT.
Import process Log Files
In the process of importing a JT file, the following log files are generated by the JT Import plug-in.

The process log and error messages are recorded in a `.err` file located in the CATIA 3DEXPERIENCE CATReport directory. The file is named after the selected input file name. e.g. if the file Mypart.jt were selected, the log file names would be `%CATReport\Mypart.err` and `%CATReport\Mypart.rpt`.

%TS_TEMP_DIR\<input_file_name>_viewer.log
%TS_TEMP_DIR\<input_file_name>_viewer_screen_output.log
%CATReport\<input_file_name>.err

Where `<input_file_name>` - is the base file name of the imported file.

CATIA 3DEXPERIENCE Environment Files

As part of the Theorem installation process, a set of CATIA 3DEXPERIENCE environment files are created which are subsequently used in the launch of 3DEXPERIENCE and CATUtil sessions to support the Theorem partner plug-ins for JT import and export.

A ‘CATEnv’ file is created for each installed version of CATIA 3DEXPERIENCE.

These environment files are located in

<installation_directory>\B214\win_b64\Theorem_Multi-CAD_JT_CATIAV6R2013x.txt
<installation_directory>\B216\win_b64\Theorem_Multi-CAD_JT_CATIAV6R2015x.txt
<installation_directory>\B417\win_b64\Theorem_Multi-CAD_JT_CATIAV6R2016x.txt
<installation_directory>\B418\win_b64\Theorem_Multi-CAD_JT_CATIAV6R2017x.txt
Appendix A – JT Configuration File

Introduction
A configuration file contains the settings for your translations. The configuration file can be specified using the command line option --config or -z.

The configuration file name can be defined in the interactive User Preferences/Tools Options form, which is in turn defined in the
TS_INST\data\jt\jt_mcad_options_configuration.txt file (where TS_INST = Installed directory).

By default this is set to TS_INST\etc\tessCATIAV6MultiCAD.config

Alternatively, if in batch mode, with the environment variable
TS_IGNORE_JT_CATSETTINGS=1 set, the interactive User Preferences/Tools Options form will be ignored, and the content of the TS_INST\data\jt\xcad_jt_opts.txt file will be used.

In this file, the default is also defined as
-z "%TS_INST%\etc\tessCATIAV6MultiCAD.config"

The JT configuration file contains various sections, each containing different settings based on the section.

The Setup Section
The setup options in the configuration file define how your files are translated. The setup section is the first part of the configuration file and contains a series of standard translator options.

To edit setup options

1. Open an existing configuration file with a text editor.
2. Edit the configuration file options listed in the table below.
3. Save the configuration with a .config extension

<table>
<thead>
<tr>
<th>Option name</th>
<th>Keywords</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAITranslator</td>
<td>EAITranslator {</td>
<td>EAITranslator {</td>
</tr>
<tr>
<td>OutputDirectory</td>
<td>&quot;path to directory&quot;</td>
<td>OutputDirectory = &quot;/home/&lt;user&gt;&quot;</td>
</tr>
<tr>
<td>CommonPartsPath</td>
<td>&quot;path to directory&quot;</td>
<td>CommonPartsPath= &quot;/myaccount/jtparts/&quot;</td>
</tr>
<tr>
<td>chordalOption</td>
<td>&quot;RELATIVE&quot;</td>
<td>chordalOption = &quot;RELATIVE&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;ABSOLUTE&quot;</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>structureOption</td>
<td>&quot;PER_PART&quot; &quot;MONOLITHIC&quot; &quot;FULL_SHATTER&quot;</td>
<td>structureOption = &quot;MONOLITHIC&quot;</td>
</tr>
<tr>
<td>WriteWhichFiles</td>
<td>&quot;ALL&quot; &quot;ASSEMBLY_ONLY&quot; &quot;PARTS_ONLY&quot;</td>
<td>WriteWhichFiles = &quot;ALL&quot;</td>
</tr>
<tr>
<td>compression</td>
<td>true TRUE false FALSE</td>
<td>compression = true</td>
</tr>
<tr>
<td>triStripOpt</td>
<td>true TRUE false FALSE</td>
<td>triStripOpt = false</td>
</tr>
<tr>
<td>seamSewing</td>
<td>true TRUE false FALSE</td>
<td>seamSewing = true</td>
</tr>
<tr>
<td>seamSewingTol</td>
<td>any integer</td>
<td>seamSewingTol = 0.001</td>
</tr>
<tr>
<td>includeBrep</td>
<td>true TRUE false FALSE</td>
<td>includeBrep = false</td>
</tr>
<tr>
<td>brepPrecision</td>
<td>&quot;SINGLE&quot; &quot;DOUBLE&quot;</td>
<td>brepPrecision = &quot;SINGLE&quot;</td>
</tr>
<tr>
<td>autoNameSanitize</td>
<td>true TRUE</td>
<td>autoNameSanitize = true</td>
</tr>
</tbody>
</table>

*Note: Not available for Unigraphics.*
<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>updateChangedPartsOnly</td>
<td>true</td>
<td><code>updateChangedPartsOnly = false</code></td>
</tr>
<tr>
<td></td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>verboseReporting</td>
<td>true</td>
<td><code>verboseReporting = false</code></td>
</tr>
<tr>
<td></td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>writeAsciiAssembly</td>
<td>true</td>
<td><code>writeAsciiAssembly = false</code></td>
</tr>
<tr>
<td></td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>singlePartsNoAssem</td>
<td>true</td>
<td><code>singlePartsNoAssem = false</code></td>
</tr>
<tr>
<td></td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>smartLODgeneration</td>
<td>true</td>
<td><code>smartLODgeneration = true</code></td>
</tr>
<tr>
<td></td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>autoLowLODgeneration</td>
<td>true</td>
<td><code>autoLowLODgeneration = true</code></td>
</tr>
<tr>
<td></td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>numLODs</td>
<td>any integer</td>
<td><code>numLODs = 3</code></td>
</tr>
</tbody>
</table>
| close brace           | )     | ) }
The Level of Detail Section
The level of detail section of the configuration file contains the tessellation and simplification information for each level of detail in the file.

This section consists of several sets of level of detail (LOD) information, and the number of these sets depends on the number you specified on the numLODs line in the configuration file.

To edit level of detail options

1. Open an existing configuration file in a text editor.
2. Edit the configuration file options listed below.
3. Save the configuration with a .config extension

<table>
<thead>
<tr>
<th>Option name</th>
<th>Keywords</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOD</td>
<td>LOD &quot;lod number&quot; {</td>
<td>LOD &quot;1&quot; {</td>
</tr>
<tr>
<td>Level</td>
<td>any integer</td>
<td>Level = 1</td>
</tr>
<tr>
<td>Chordal</td>
<td>any number</td>
<td>Chordal = 0.001</td>
</tr>
<tr>
<td>Angular</td>
<td>any number</td>
<td>Angular = 25</td>
</tr>
<tr>
<td>Length</td>
<td>any number</td>
<td>Length = 1</td>
</tr>
<tr>
<td>FeatureSuppression</td>
<td>any integer</td>
<td>FeatureSuppression = 0</td>
</tr>
<tr>
<td>Simplify</td>
<td>any number</td>
<td>Simplify = 0.60</td>
</tr>
<tr>
<td>close brace</td>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>

The Filter Section
The filter section of the configuration file contains the filename and metadata filtering information. Edit this section if you want to change how the translator sanitizes filenames and filters metadata keys.

To edit filter options

1. Open an existing configuration file with a text editor.
2. Edit the configuration file options from the table below.
3. Save the configuration with a .config extension

<table>
<thead>
<tr>
<th>Option name</th>
<th>Keywords</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Filter {</td>
<td>Filter {</td>
</tr>
<tr>
<td>FilenameSanitizeSet</td>
<td>&quot;string of characters&quot;</td>
<td>FilenameSanitizeSet = &quot;abc123.&quot;</td>
</tr>
</tbody>
</table>
The Metadata section
The metadata section sets which metadata to attach to all parts, assemblies and nodes of the model.

Note: Be sure to add these options to the configuration file in pairs: one line to define the metadata key and one line to define the metadata value.

To edit metadata options
1. Open an existing configuration file (.CONFIG) in a text editor.
2. Edit the configuration file options shown in the table below.
3. Save the configuration with a .config extension

<table>
<thead>
<tr>
<th>Option name</th>
<th>Keywords</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>Metadata {</td>
<td>Metadata {</td>
</tr>
<tr>
<td>AddToParts</td>
<td>&quot;string of characters&quot;</td>
<td>AddToParts = &quot;&lt;metadata key&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AddToParts = &quot;&lt;metadata value&gt;&quot;</td>
</tr>
<tr>
<td>AddToAssemblies</td>
<td>&quot;string of characters&quot;</td>
<td>AddToAssemblies = &quot;&lt;metadata key&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AddToAssemblies = &quot;&lt;metadata value&gt;&quot;</td>
</tr>
<tr>
<td>AddToAllNodes</td>
<td>&quot;string of characters&quot;</td>
<td>AddToAllNodes = &quot;&lt;metadata key&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AddToAllNodes = &quot;&lt;metadata value&gt;&quot;</td>
</tr>
<tr>
<td>close brace</td>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>
Appendix B – Large Assembly Processing (LAP)

Overview

The export of large assemblies from 3DEXPERIENCE V6 CATIA to JT may be handled using the default process, or a new Large Assembly Processing method.

This new process is as follows:

The assembly is read from 3DEXPERIENCE V6 using the MultiCAD interfaces as normal, but the assembly structure only is directly converted to a main output jt file.

The reference to the geometry for each individual component part node in the assembly is written to separate .xml files. This part of the process takes minimal time and processing resource.

As each of the individual .xml files are created, an entry is made into a batch processing file to allow subsequent conversion of the geometry data into the output JT files required for the complete assembly.

LAP Options

The following option support has been added into the V6 > JT product to support Large Assembly Processing (LAP). This can be modified in the

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>struct_read (mandatory)</td>
<td>This option causes only the CATIA assembly structure to be written to the specified output jt file.</td>
</tr>
<tr>
<td>large_assy_process (mandatory)</td>
<td>This option invokes the creation of separate intermediate data files representing each assembly ‘leaf node’ (component/part) containing the part geometry. This option should always be used with the ‘struct_read’ option. A batch command file (.bat) is also created and this contains a sequence of individual commands to convert the intermediate data files into the required jt files representing the part/component geometry. The default name for the generated batch file is &lt;output_file_path&gt;.bat, e.g if the output file name was C:\parts\jt\assembly1.jt, the batch file name would be C:\parts\jt\assembly1.bat.</td>
</tr>
<tr>
<td>write_assembly_script (optional)</td>
<td>This option allows the user to specify a non-default file name path for the batch command file generated by the large_assy_process option.</td>
</tr>
</tbody>
</table>
### autorun (optional)

This option will cause the batch command script to be automatically invoked when the main conversion process ends.

### zpart (optional)

This option specifies the name of a JT write config file to be used in the batch file conversions for creating the jt files representing the part geometry. This will override the –z option used for the main assembly conversion.

---

### Interactive Operation

The Large Assembly Processing facility can be used in interactive mode by including the required options in the %TS_INST%\data\jt\xcad_jt_opts.txt file.

###DataExchangePLMBatch Operation

The Large Assembly Processing facility can be used in DataExchangePLMBatch mode.

### Batch Mode Operation

The options for Large Assembly Processing can be used as command line options on the new catia6_jt.cmd command line (see Appendix C).

For Interactive or DataExchangePLMBatch Large Assembly Processing, set the following in the relevant environment file:

```
e.g. C:\Program Files\Theorem\20.2\B214\win_b64\CATEnvTheorem_Multi-CAD_JT_CATIAV6R2013x.txt
```

to suit your installation:

```plaintext
TS_V6_LAP_INPUT_TEMPLATE=C:\PPC\Products\MC6JT\master_template.xml
TS_PLM_PRODUCT_REF_TYPE_NAME=ENOSTProductReference
TS_PLM_REPRESENTATION_TYPE_NAME=ENOSTRepresentation
```
Master Template

Some non-working examples of the master_template.xml can be found in the C:\Program Files\Theorem\XX.X\Samples\3DExperience\folder of your installation. The lines highlighted in yellow should be modified with ‘your’ login details required for the process to access Enovia. The lines highlighted in green are dummy data which are replaces by the LAP process.

```xml
<?xml version="1.0" encoding="utf-8"?>

<mc6_read>
  <!-- parameters must be in this ORDER -->
  <!-- parameters only the value passed to V6 -->
  <parameters>
    <attribute name="repository" value="PLM1"/>
    <attribute name="ServerName" value="3dspace.theorem.com"/>
    <attribute name="ServerPort" value="447"/>
    <attribute name="ServerRootURI" value="3dspace"/>
    <attribute name="LoginTicket" value="REEyNzM3M0Q1REM2NDgxQzlFNzk1QzIwNjZGRDYz0DN8Um9iaW58Um9iaW58fHwWFwA=="/>
    <attribute name="PLMType" value="VPMReference"/>
    <attribute name="ReportDirectory" value="C:\TEMP\V6Export"/>
    <attribute name="LicenseData" value="LIC"/>
    <attribute name="BatchXMLFileName" value="Default"/>
  </parameters>
  <!-- arguments both key and value are passed to V6 -->
  <arguments>
    <attribute key="PLM_ExternalID" value="prd-Interfix to be defined 51097330-00069076"/>
    <attribute key="V_version" value="A"/>
  </arguments>
</mc6_read>

More information regarding this file structure is given in Appendix C
For 2015X onwards, a login ticket should be used rather than <user><password>. Details of how to create a login ticket are given in Appendix D
Appendix C – Batch Processing

As part of the development of Large Assembly Processing, a new batch processing utility has been created for 3DEXPERIENCE V6 CATIA export to jt.

This runs as a command line executable which requires the following inputs:-

<TS_INST>\bin\catia6_jt.cmd <input_file> <output_file> -z <config_file> <options>

Where:-

<TS_INST>

is the Theorem Solutions software installation directory.

<input_file>

is an xml file defining the access to a specific object in a specified 3DEXPERIENCE PLM database.

This file provides user login details (V6R2013x) or a Login Ticket (R2015x, 2015x, 2017x), to a specified Enovia repository plus a set of 3 attribute names and values which will uniquely identify the input PLM object of the conversion.

Here is an example of the xml input file for 3DEXPERIENCE V6 R2013x:-

```xml
<?xml version="1.0" encoding="utf-8"?>
<3dex_read>
<!-- parameters must be in this ORDER -->
<!-- parameters only the value passed to V6 -->
<parameters>
  <attribute name="repository" value="PLM1"/>
  <attribute name="server" value="http://my-svr1:7023/enovia"/>
  <attribute name="user" value="User"/>
  <attribute name="pw" value="User"/>
  <attribute name="SecurityContext" value="VPLMProjectLeader.Company Name.DemoDesign"/>
  <attribute name="PLMType" value="VPMReference"/>
  <attribute name="ReportDirectory" value="C:\\TEMP\\V6Export"/>
  <attribute name="LicenseData" value="LIC"/>
  <attribute name="BatchXMLFileName" value="Default"/>
</parameters>
<!-- arguments both key and value are passed to V6 -->
<arguments>
  <attribute key="PLM_ExternalID" value="prd032671"/>
  <attribute key="V_version" value="---"/>
  <attribute key="minorrevision" value="1"/>
</arguments>
</3dex_read>
```
Here is an example of the xml input file for 3DEXPERIENCE V6 R2015x:-

```xml
<?xml version="1.0" encoding="utf-8"?>

<3dex_read>
<! parameters must be in this ORDER -->
<! parameters only the value passed to V6 -->
<parameters>
<attribute name="repository" value="PLM1"/>
<attribute name="ServerName" value="3dspace.theorem.com"/>
<attribute name="ServerPort" value="447"/>
<attribute name="ServerRootURI" value="3dspace"/>
<attribute name="LoginTicket" value="REEyNzM3M0Q1REM2NDgxQzlFNzk1QzlwNjZGRDYzODN8Um9iaW58Um9iaW58fHwwfA=="/>
<attribute name="PLMType" value="VPMReference"/>
<attribute name="ReportDirectory" value="C:\TEMP\V6Export"/>
<attribute name="LicenseData" value="LIC"/>
<attribute name="BatchXMLFileName" value="Default"/>
</parameters>
<! arguments both key and value are passed to V6 -->
<arguments>
<attribute key="PLM_ExternalID" value="prd-Interfix to be defined 51097330-00023109"/>
<attribute key="V_version" value="A"/>
<attribute key="minorrevision" value="1"/>
</arguments>
</3dex_read>

-o <output_file>

is the required output JT file name.

-z <config_file>

Is the name of the JT configuration file used for controlling the output characteristics described in Appendix A.

progress_file <file name>

The path and file name for the log file e.g. C:\TEMP\progress.log
Appendix D – Creating a Login Ticket

Use a browser link (similar to the one below) to your 3DEXperience Server

https://3dspace.2017x.theorem.com:447/3dspace/common/emxNavigator.jsp

A page will appear in the browser

Login as normal, e.g.

_User_

_Password_

Select Collaboration and Approvals > Experience Configuration > Manage Login Tickets
Select the values required for your user:

User and Security Context should already be set, make sure that Ticket Type: Infinite is selected. Then click on ‘Create’

“Create” will produce the ticket:-

REEyNzM3STE1MER2NDgxQzIwNjZGNATzDAN8Um9iaW58Um9iaW58fHwwfA==

This can then be used as the ‘LoginTicket’ value in the job xml file described in Appendix C