USER GUIDE

JT - 3DEXPERIENCE CATIA V6

Release Version : 24.0

Revision: 4.0

Issued: 17/03/2022

Contents

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

Theorem have 3 main Product brands. These are:

TRANSLATE

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

See our website for more detail.

PUBLISH

The creation of documents enriched with 3D content

See our website for more detail.

VISUALIZE

Visualization for Augmented (AR), Mixed (MR) and Virtual (VR) Reality applications

See our website for more detail.

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 ○ 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 covers 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:

<installation_directory>\bin

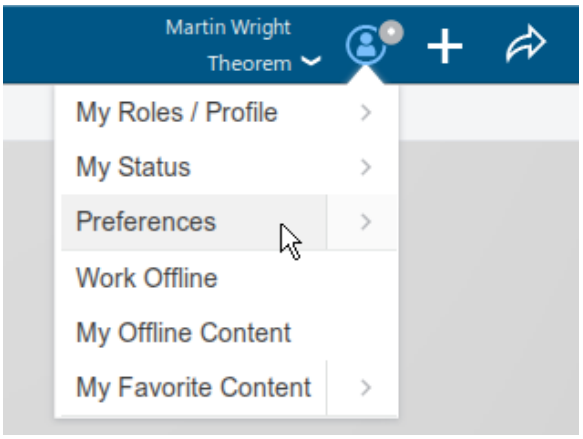
The script name is:

start_3DEXPERIENCE_Theorem_Multi-CAD_JT_CATIAV6R<version>.cmd

(where <version> should be substituted for the version of CATIA 3DEXPERIENCE that you have installed – e.g. 2017x, 2018x, 2019x, 2020x):

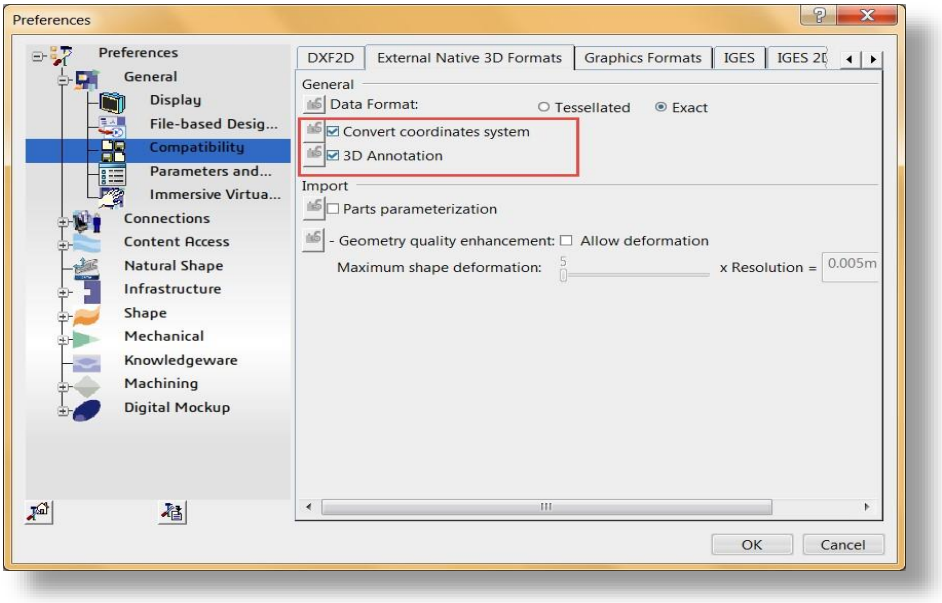
Theorem Interactive Conversion Settings

The 3DEXPERIENCE interface does not currently require the user to apply any Creo View specific settings for the translation. There are some general settings that should be checked if required (e.g. for PMI conversion.)For R2015x and later revisions these are accessed through Preferences>Legacy Preferences>General> Compatibility > External Native 3D Formats :

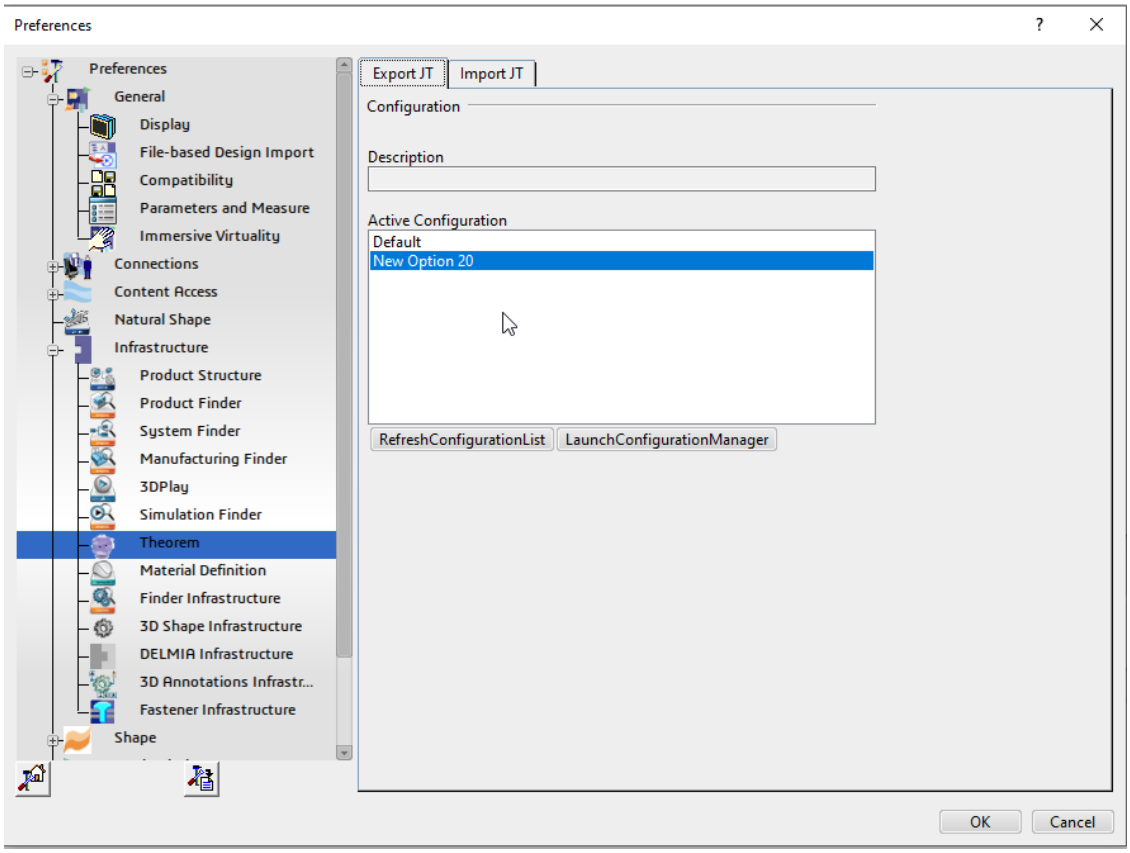


3DEXPERIENCE 2018x

3DEXPERIENCE 2019x onwards



This page is a standard Dassault Page that sets the preferred mode of conversion (in this case Exact), and also the general options “Convert coordinate systems” and “3D Annotation”. Two dedicated tabs under ***‘Infrastructure>Theorem’*** allow the user access to Theorem Configurations for import and export is available under the ***‘Infrastructure’*** settings and is presented as follows:-

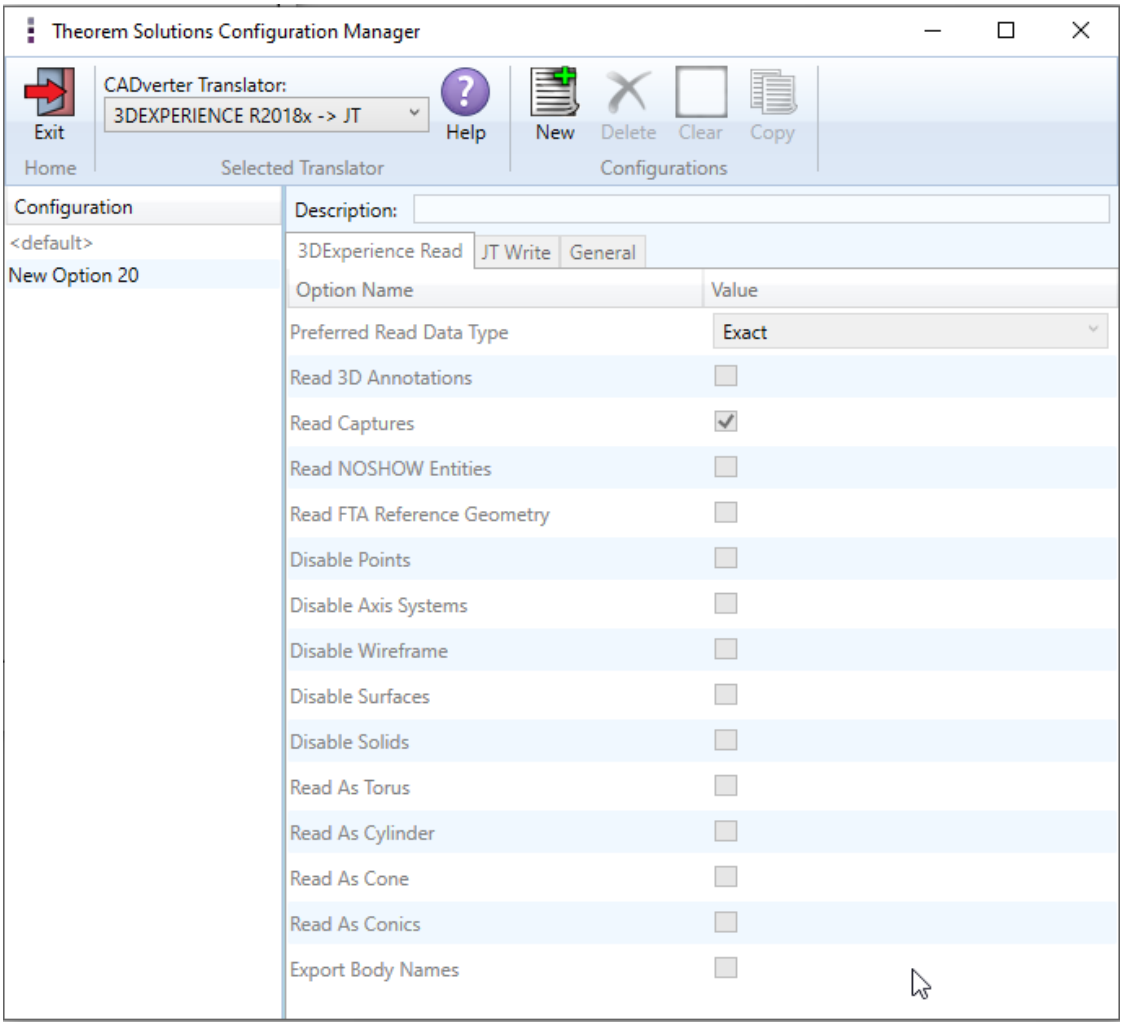


From this Panel, the user can select a predefined configuration, these are set from within the Configuration Manager. You can also launch and refresh the Configuration Manager.

Configuration Manager

3DExperience to JT export

3DExperience Read



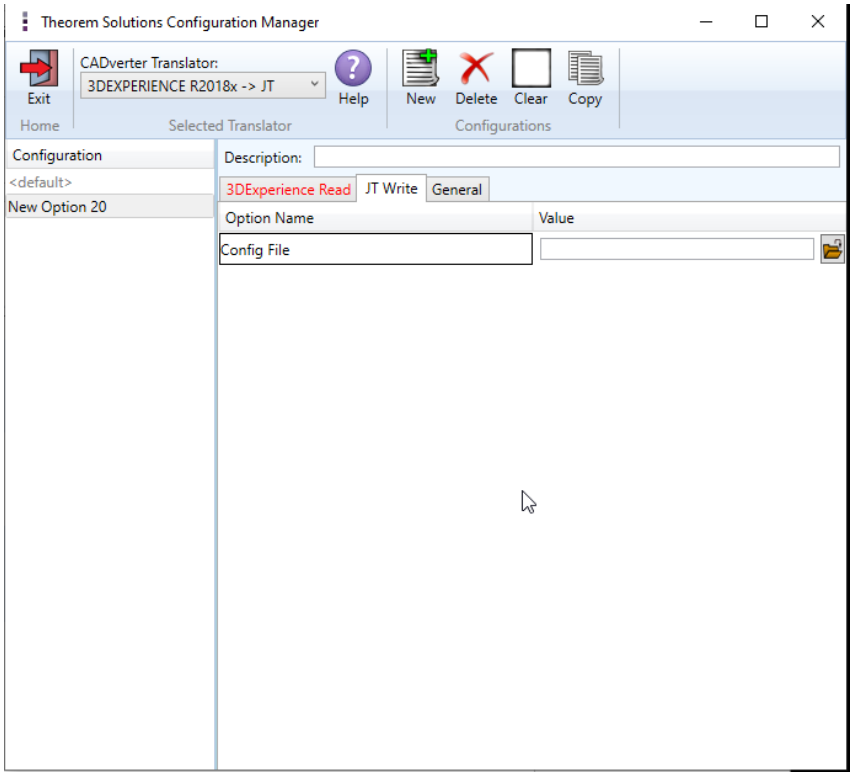
Each of these options is described below:

Option	Description
Preferred Read Data Type	The setting options are EXACT (default) or TESSELATED • Command Line Syntax: *read_tess*

Read 3D Annotations	Enables PMI data read from the V5 file. *(Default is OFF).* o Command Line Syntax: ▪ *dont_read_pmi – to turn off* ▪ *read_pmi- to turn on* ***Note!*** When *'read_pmi'* is enabled it also enables the '*fill_pmi_arrows'*, *'fill_pmi_text'* and '*pmi_filled_text'* options. These can be overridden by setting the Advanced arguments: *'dont_fill_pmi_arrows'* and/or
Read Captures	A secondary argument to 'Read PMI' and allows the control over whether captures are read as part of the process. Default is ON when 'Read PMI' is marked as ON. • Command Line Syntax: *read_captures - default* *dont_read_captures – to turn off*
Read NOSHOW entities	Read any entities that are in NOSHOW. Default is to not read NOSHOW entities • Command Line Syntax:
Read FTA Reference Geometry	Enables reading of FTA Reference Geometry *(Default is Off)* • Command Line Syntax: *read_geometry – to turn on*
Disable Points	Switches off Point processing • Command Line Syntax: *disable_points*
Disable Axis Systems	Switches off Axis System processing • Command Line Syntax: *disable_axes*
Disable wireframe	Switches off Wireframe processing • Command Line Syntax: *disable_wireframe*
Disable Surfaces	Switches off Surface processing • Command Line Syntax: disable_surfaces
Disable Solids	Switches off Solids processing • Command Line Syntax: *disable_solids*

Read as Torus	Read Toroidal surfaces in analytical form (default is NURBS) • Command Line Syntax: *read_torus*
Read as Cylinder	Read Cylindrical surfaces in analytical form (default is NURBS) • Command Line Syntax: *read_cylinder*
Read as Cone	Read Cone surfaces in analytical form (default is NURBS) • Command Line Syntax: *read_cone*
Read as Conic	Read surfaces generated from a Conic in analytical form (default is NURBS) • Command Line Syntax: *read_conics*
Export Body Names	Maintains body names for parts that consist of multiple bodies. • Command Line Syntax: *body_names*

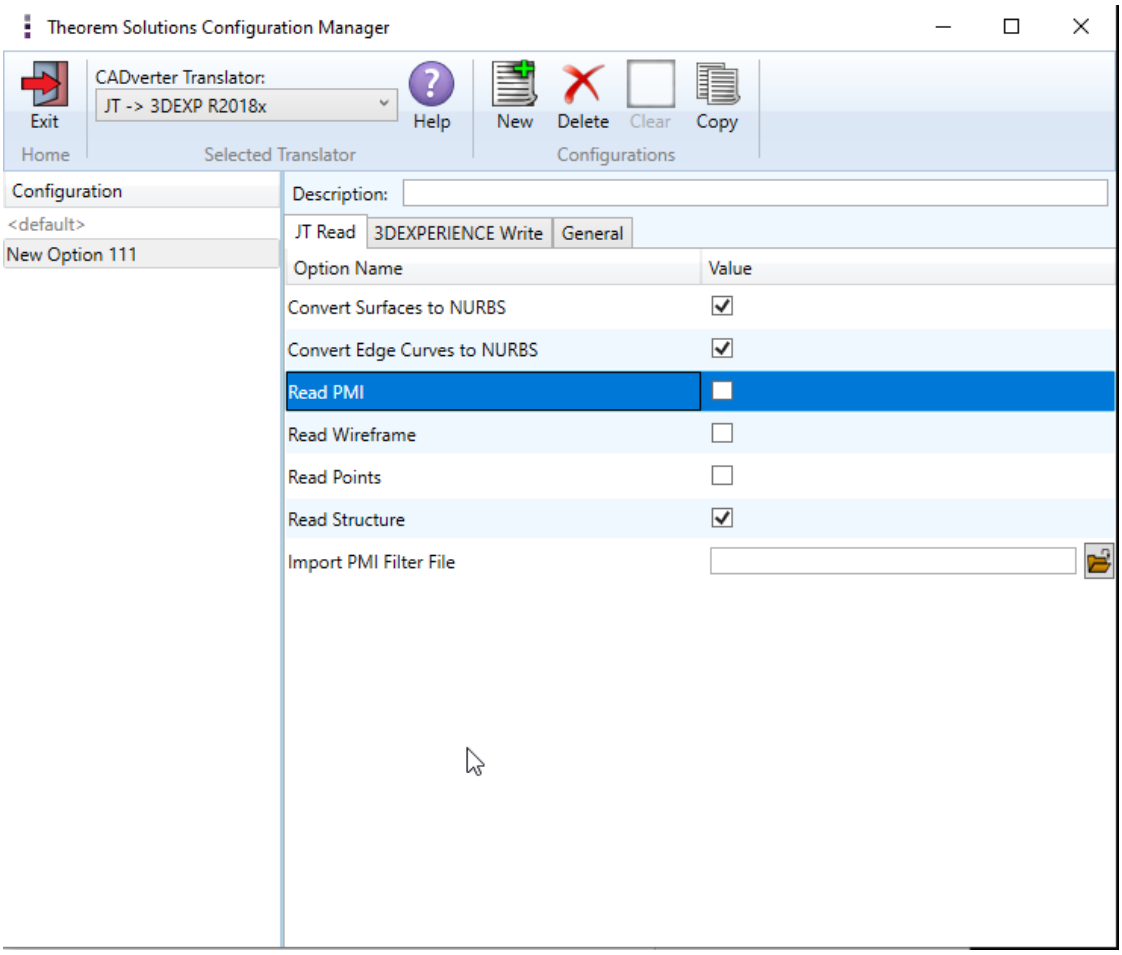
JT Write



Enter the name of the JT config file. The default is "<TS_INST>/etc/tess.config" (see Appendix A for config file options)

JT to 3DExperience import

JT Read

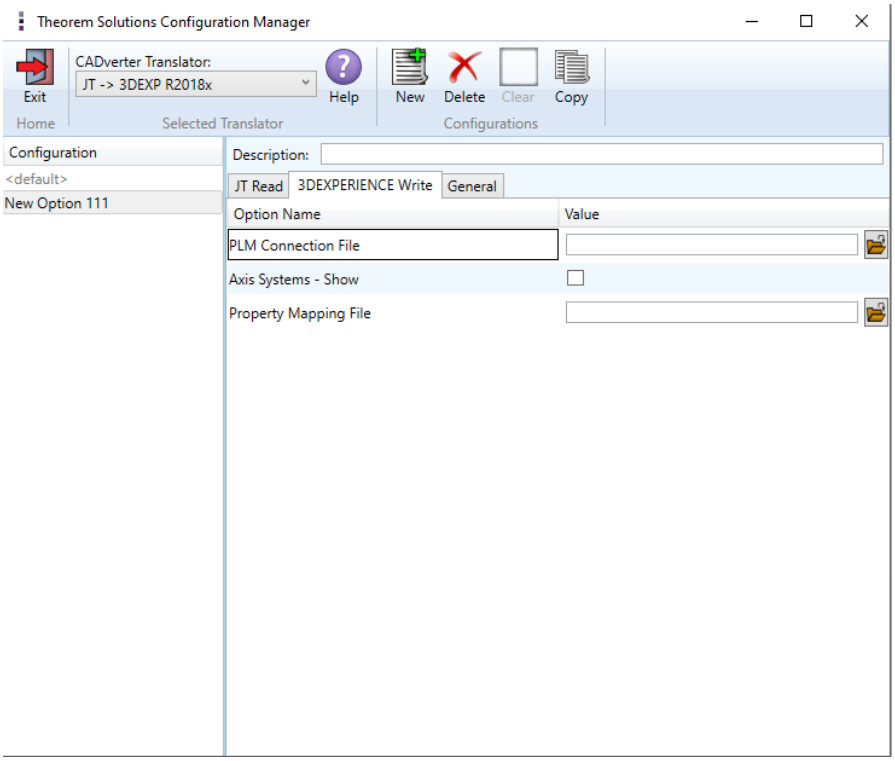


Each of these options is described below.

Option	Description
Convert Surfaces to NURBS	Read XT Brep surfaces as NURBS surfaces (else read in native form). Default is ON. ○ Command Line Syntax ▪ *noprep – to turn off*

Convert Curves to NURBS	Read XT Brep edge curves as NURBS curves (else read in native form. Default is ON. ○ Command Line Syntax ▪ *rd_native_edge – to turn off*
Read PMI	Reads 3D PMI. Default is OFF. ○ Command Line Syntax ▪ *read_pmi dim2_pmi*
Read Wireframe	Read JT wireframe data. Default is OFF. ○ Command Line Syntax ▪ *read_wire_frame*
Read Points	Read JT point data. Default is OFF. ○ Command Line Syntax ▪ *read_points*
Read Structure	Read assembly tree structure. Default is ON
Import PMI filter file	Supply PMI filter file. Default is "<TS_INST>/data/jt_pmi_filter.txt" ○ Command Line Syntax pmi_filter_file "file name"

3DExperience write



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>

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.

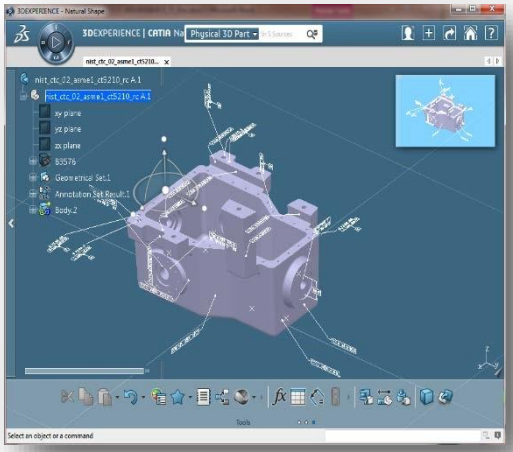
This facility was introduced to reduce processing time due to large numbers of 'redundant' PMI entities in a JT file.

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.*

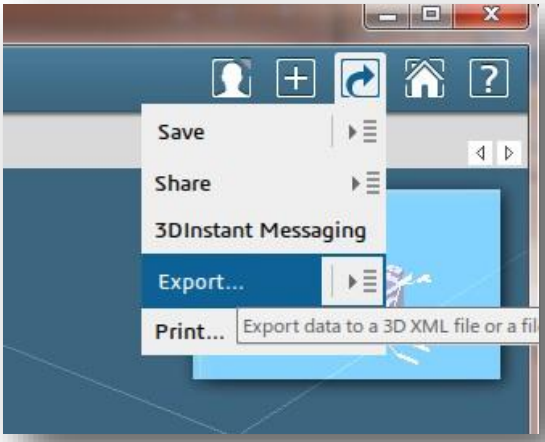
Default settings that can control the JT Import plugin, are also read from the data file *%TS_INST%\data\jt\jt_xcad_opts.txt.*

Theorem Export

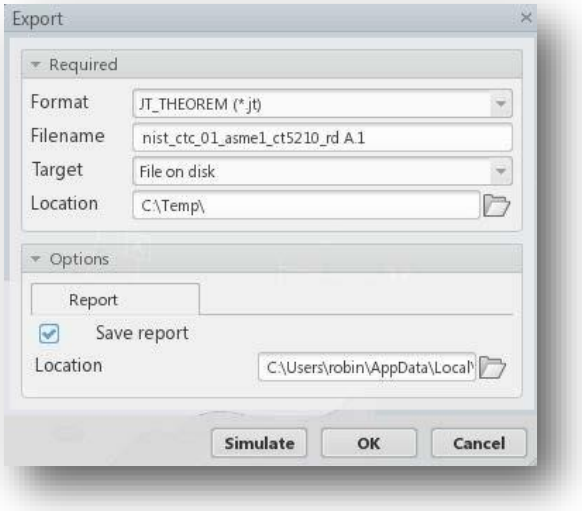
Once CATIA 3DEXPERIENCE has been started it is possible for an opened product or representation to be exported to JT.



The user selects the menu item as shown here:-



This menu action will then present the user with an Export dialog box similar to this:-



The “Format” selection should be set to JT_THEOREM (*.jt).

The ***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 a template JT file (named by default as template_empty.jt, template_hidden.jt 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%\B422\win_b64\CATEnv\Theorem_Multi-CAD_JT_CATIAV6R<ver>.txt file, typically as follows:

TS_JT_TEMPLATE_FILE=C:\Program Files\Theorem\20.2\data\jt\template.jt

TS_JT_TEMPLATE_FILE_EMPTY=C:\Program Files\Theorem\20.2\data\jt\template_empty.jt

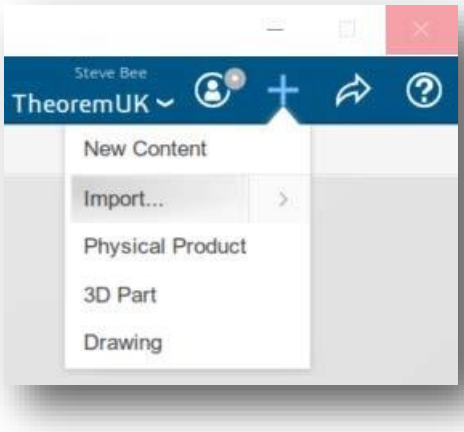
TS_JT_TEMPLATE_FILE_HIDDEN=C:\Program Files\Theorem\20.2\data\jt\template_hidden.jt

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

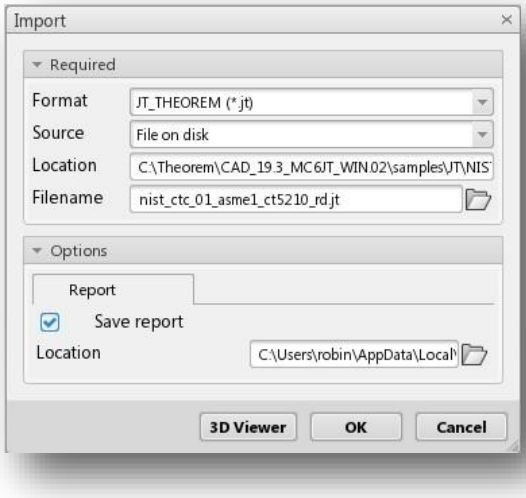
Theorem Import

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:



This launches the Theorem Import Menu:



Select the JT_THEOREM file type to be opened (.jt/.plmxml/.stpx). After choosing the file to import, selecting ***OK*** on the Import Menu will start the translation and the JT data will be 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. 2017x, 2018x, 2019x, 2020x)

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.

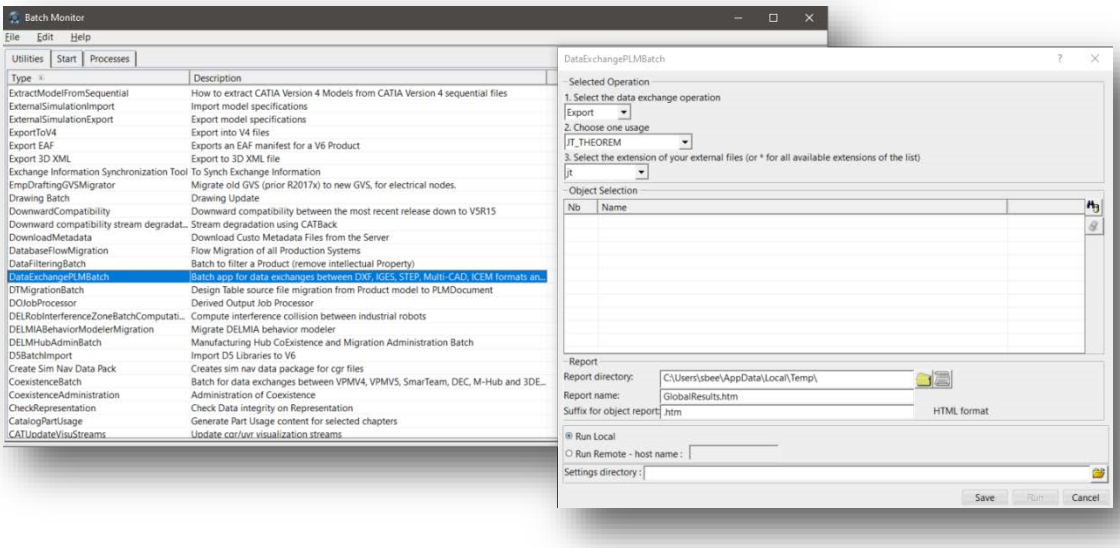
In some cases, it may be desirable to ignore the interactive settings while running in batch mode.

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

TS_IGNORE_JT_CATSETTINGS=1

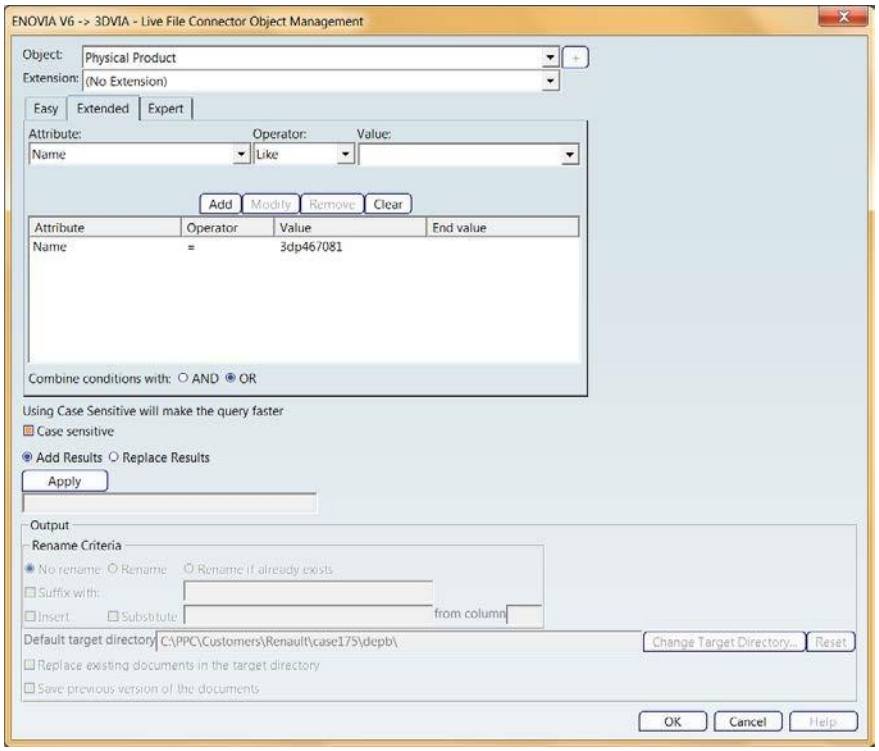
Batch Export to JT

Here is an example showing the appropriate settings in preparation for a database selection and an export



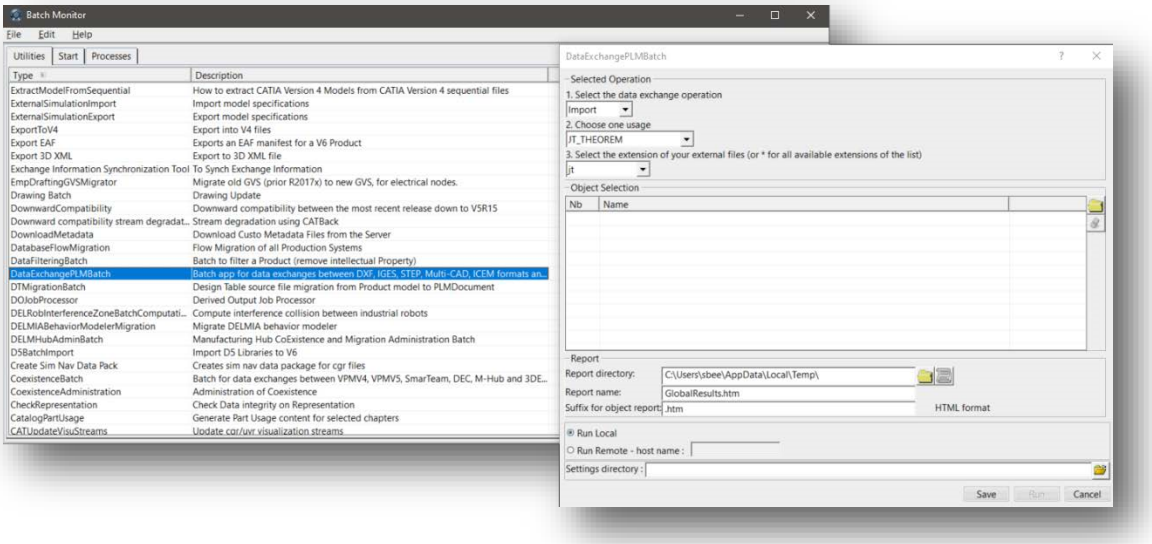
Note selection of “Export” operation, usage “JT_THEOREM” and extension “jt”.

The Representation or Product for export can be found using the standard Enovia V6 selection tools  similar to that shown below



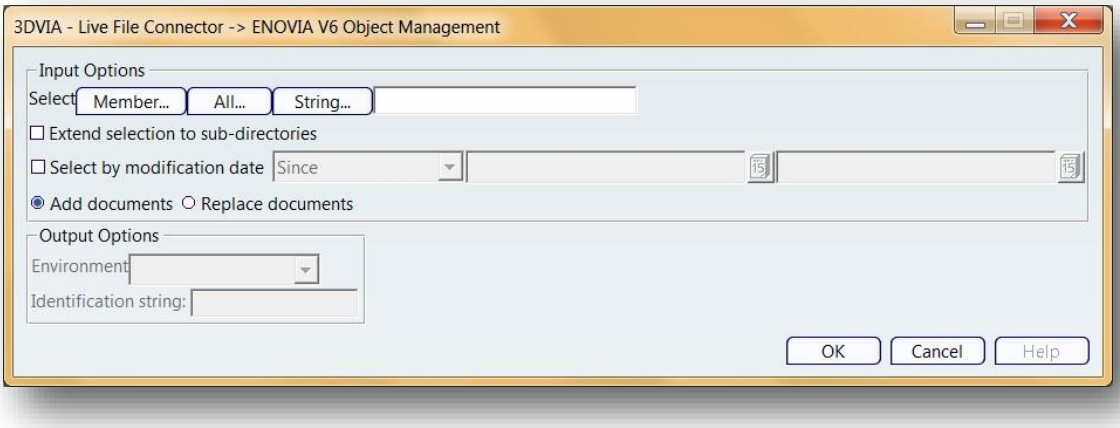
Batch Import from JT

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.



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.

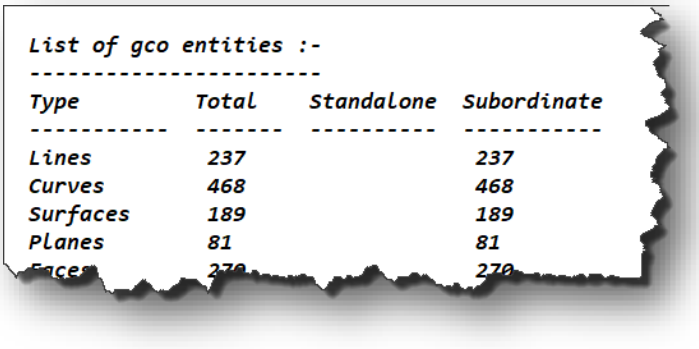
The process log and error messages are recorded in a '.err' file located in the CATIA 3DEXPERIENCE CATReport directory. The file is named after the active CATIA component.

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. Normally a list of entities.



```
List of gco entities :-
-----------------------
Type          Total     Standalone  Subordinate
-----------   -------   ----------  -----------
Lines          237                   237
Curves         468                   468
Surfaces       189                   189
Planes         81                    81
Faces          270                   270
```

%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. The status of the translation can be found here

%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 and contains additional information such as modifiers and options used.

Import process Log Files

In the process of importing a JT file, the following log files are generated by the JT Import plug-in.

- nist_ftc_08_asme1_ct5240_rc.err – *gives the full processing list of errors, warnings and information*
- nist_ftc_08_asme1_ct5240_rc.log.rpt – gives a short list of the entities created and failed
- nist_ftc_08_asme1_ct5240_rc.log.summary – gives the times for start and finish and the status message code (these can be customised)
- nist_ftc_08_asme1_ct5240_rc.log – *gives a single file with the data from all three logs*

The process log and error messages are, by default, located in the CATIA 3DEXPERIENCE CATReport directory. The files is named after the selected input file name. e.g. Mypart.jt would produce the log file names *Mypart.err, Mypart.rpt etc.*

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.

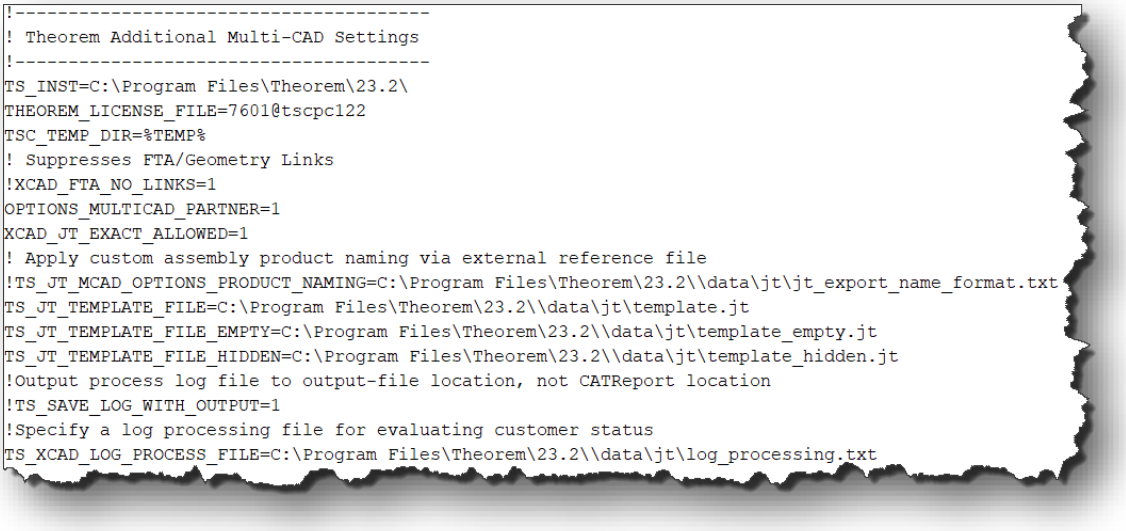
A 'CATEnv' file is created for each installed version of CATIA 3DEXPERIENCE.

These environment files are located in the 3DExperience revision specific folder e.g.

<installation_directory>\B421\win_b64\Theorem_Multi-CAD_JT_CATIAV6R2019x.txt

<installation_directory>\B422\win_b64\Theorem_Multi-CAD_JT_CATIAV6R2020x.txt

These files consist of the current 3DExperience settings with the required Theorem settings appended at the bottom.



```
!---------------------------------------
! Theorem Additional Multi-CAD Settings
!---------------------------------------
TS_INST=C:\Program Files\Theorem\23.2\
THEOREM_LICENSE_FILE=7601@tscpc122
TSC_TEMP_DIR=%TEMP%
! Suppresses FTA/Geometry Links
!XCAD_FTA_NO_LINKS=1
OPTIONS_MULTICAD_PARTNER=1
XCAD_JT_EXACT_ALLOWED=1
! Apply custom assembly product naming via external reference file
!TS_JT_MCAD_OPTIONS_PRODUCT_NAMING=C:\Program Files\Theorem\23.2\\data\jt\jt_export_name_format.txt
TS_JT_TEMPLATE_FILE=C:\Program Files\Theorem\23.2\\data\jt\template.jt
TS_JT_TEMPLATE_FILE_EMPTY=C:\Program Files\Theorem\23.2\\data\jt\template_empty.jt
TS_JT_TEMPLATE_FILE_HIDDEN=C:\Program Files\Theorem\23.2\\data\jt\template_hidden.jt
!Output process log file to output-file location, not CATReport location
!TS_SAVE_LOG_WITH_OUTPUT=1
!Specify a log processing file for evaluating customer status
TS_XCAD_LOG_PROCESS_FILE=C:\Program Files\Theorem\23.2\\data\jt\log_processing.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.

For interactive or batch users the configuration file name can be defined in the Preferences/Legacy Preferences\Infrastructure\Theorem form, which in turn points to the TS_INST\data\jt\jt_mcad_options_configuration.txt file (where TS_INST = the Theorem Installation folder).

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 Preferences/Legacy Preferences\Infrastructure\Theorem 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

Option name	Keywords	Example
EAITranslator	`EAITranslator {`	`EAITranslator {`
OutputDirectory	`"path to directory"`	`OutputDirectory = "/home/<user>/"`
CommonPartsPath	`"path to directory"`	`CommonPartsPath= "/myaccount/jtparts/"`
chordalOption	`"RELATIVE"`	`chordalOption = "RELATIVE"`
	`"ABSOLUTE"`	

structureOption	`"PER_PART"`
	`"MONOLITHIC"`
	`"FULL_SHATTER"`
		`structureOption = "MONOLITHIC"`

WriteWhichFiles	"ALL" "ASSEMBLY_ONLY" "PARTS_ONLY"	WriteWhichFiles = "ALL"
compression	true TRUE false FALSE	compression = true
triStripOpt	true TRUE false FALSE	triStripOpt = false
seamSewing **Note: Not available for Unigraphics.**	true TRUE false FALSE	seamSewing = true
seamSewingTol	*any integer*	seamSewingTol = 0.001
includeBrep	true TRUE false FALSE	includeBrep = false
brepPrecision	"SINGLE" "DOUBLE"	brepPrecision = "SINGLE"
autoNameSanitize	true TRUE	autoNameSanitize = true

	false FALSE	
updateChangedPartsOnly	true TRUE false FALSE	updateChangedPartsOnly = false
verboseReporting	true TRUE false FALSE	verboseReporting = false
writeAsciiAssembly	true TRUE false FALSE	writeAsciiAssembly = false
singlePartsNoAssem	true TRUE false FALSE	singlePartsNoAssem = false
smartLODgeneration	true TRUE false FALSE	smartLODgeneration = true
autoLowLODgeneration	true TRUE false FALSE	autoLowLODgeneration = true
numLODs	*any integer*	numLODs = 3
close brace		

} }

The Level of Detail (LOD) 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

Option name	Keywords	Example
LOD	LOD "*lod number*" {	LOD "1" {
Level	*any integer*	Level = 1
Chordal	*any number*	Chordal = 0.001
Angular	*any number*	Angular = 25
Length	*any number*	Length = 1
FeatureSuppression	*any integer*	FeatureSuppression = 0
Simplify	*any number*	Simplify = 0.60
close brace		
	}	}

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

Option name	Keywords	Example
Filter	`Filter {`	`Filter {`
FilenameSanitizeSet	*`"string of characters"`*	`FilenameSanitizeSet =`
FilenameSanitizeSetAdd	*`"string of characters"`*	`FilenameSanitizeSetAdd = "4l"`
FilenameSanitizeSetDelete	*`"string of characters"`*	`FilenameSanitizeSetDelete = "c"`
MetadataKey	*`"string of characters"`*	`MetadataKey = "metadata key to exclude"`
close brace		
		`"abc123."`
	`}`	`}`

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

Option name	Keywords	Example

Metadata	Metadata {	Metadata {
AddToParts	*"string of characters"*	AddToParts = "<metadata key>" AddToParts = "<metadata value>"
AddToAssemblies	*"string of characters"*	AddToAssemblies = "<metadata key>" AddToAssemblies = "<metadata value>"
AddToAllNodes	*"string of characters"*	AddToAllNodes = "<metadata key>" AddToAllNodes = "<metadata value>"
close brace		
	}	}

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

Option	**Description**

struct_read **(mandatory)**	This option causes only the CATIA assembly structure to be written to the specified output jt file.
large_assy_process **(mandatory)**	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 <output_file_path>.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.
write_assembly_script **(optional)**	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.
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_MultiCAD_JT_CATIAV6R2013x.txt

to suit your installation

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 replaced by the LAP process.

```
<?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="REEyNzM3M0Q1REM2NDgxQzlFNzk1QzIwNjZGRDYz0DN8Um9iaW58Um9iaW58fHw
wfA==/>
   <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"/>
   <attribute key="minorrevision" value="1"/>
  </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 R2015x:-

```
<?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="REEyNzM3M0Q1REM2NDgxQzlFNzk1QzIwNjZGRDYzODN8Um9iaW58Um9ia
W58fHwwfA==="/>
   <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
5109733000023109"/>
   <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>

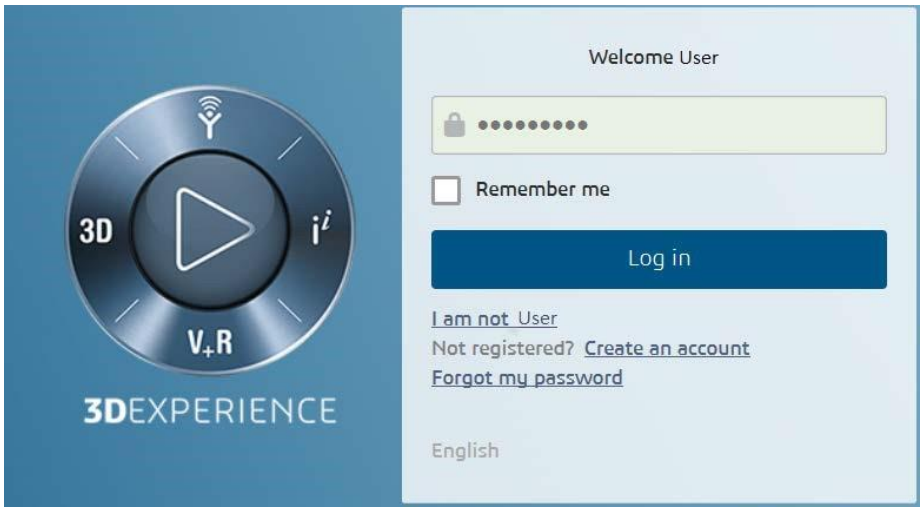
The path and file name for the log file e.g. C:\TEMP\progress.log

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

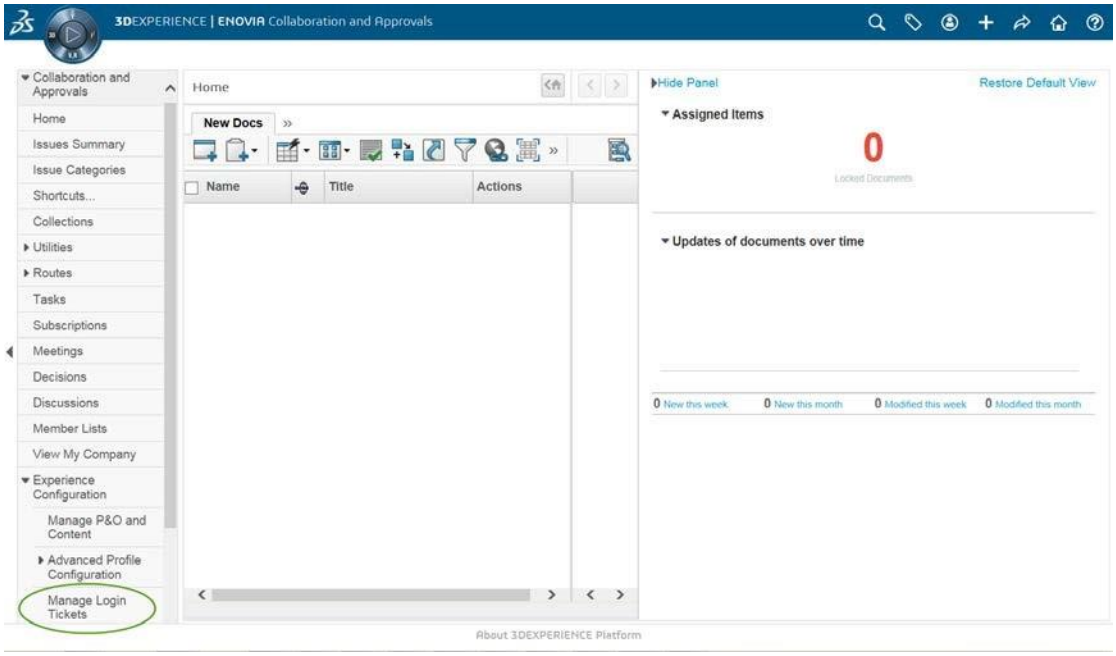


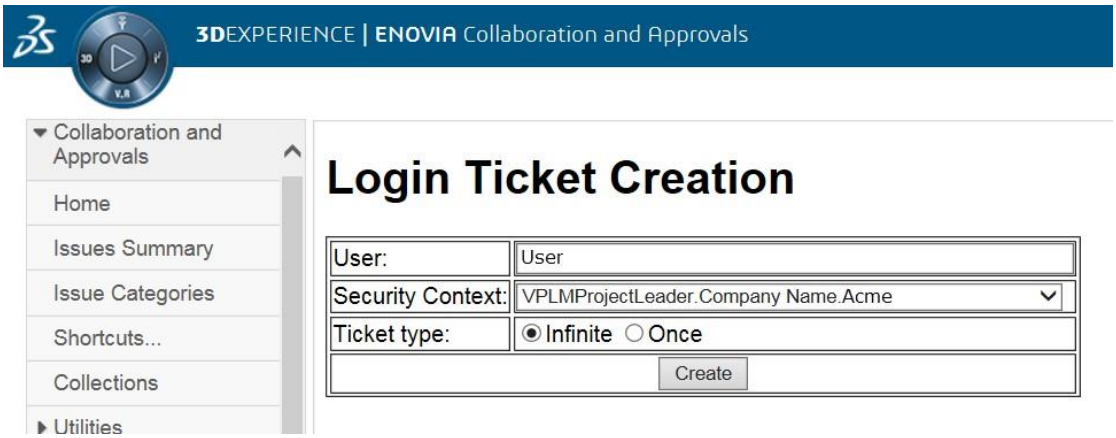
Login as normal, e.g.

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:-

REEyNzM3STE1MER2NDgxQzlFNzk1QzIwNjZGNATzDAN8Um9iaW58Um9iaW58fHwwfA==

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