



TRANSLATE for 3DEXPERIENCE <> JT



USER GUIDE

Version 25.3

Revision: 1.0
Issued: 15/11/2022

Contents

Overview of Translate	3
About Theorem	3
Theorem's Product Suite	4
The 3DEXPERIENCE to JT Bi-directional Translator	5
Primary Product Features	5
Primary Product benefits?	6
Getting Started	7
Documentation & Installation Media	7
Using the Product.....	9
Translating Interactively from within 3DEXPERIENCE	9
Theorem Interactive Conversion Settings.....	10
Configuration Manager	13
3DEXPERIENCE to JT Export.....	13
JT to 3DEXPERIENCE Import	17
Property Mapping Files.....	20
PMI Type Filter	22
Theorem Export.....	23
JT Template Files.....	25
Theorem Import	26
Translating in Batch using CATUtil – DataExchangePLMBatch.....	27
Batch Export to JT.....	28
Batch Import from JT	30
Log File Generation.....	32
Export process Log Files	32
Import process Log Files	33
CATIA 3DEXPERIENCE Environment Files	34
Appendix A – JT Configuration File	35
Introduction.....	35
The Setup Section.....	36
The Level of Detail (LOD) Section	39
The Filter Section.....	40
The Metadata section.....	41
Appendix B – Large Assembly Processing (LAP)	42
Overview.....	42

LAP Options	43
Interactive Operation	44
DataExchangePLMBatch Operation	44
Batch Mode Operation	44
Master Template	45
Appendix C – Batch Processing.....	46
Appendix D – Creating a Login Ticket.....	48

Overview of Translate

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 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 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, wireframe, 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
- 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 facettted representation, a JTBrep definition or an XTbrep definition. As standard a facettted 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 industrial use Theorem's product robustness and quality is well proven, reducing your business risk

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

Getting Started

Documentation & Installation Media

The latest copy of the User Guide documentation can be found on our web site at:

<http://www.theorem.com/Documentation>

Each product has a specific link that provides user documentation in the form of PDF and Tutorials.

The latest copy of Theorem software can be found via the link above and by searching for the specific product.

Latest Release: Version V24.0

- [Product Release Notes](#)
- [User Guide](#)
- [Installation and Configuration Guide](#)

JT File Analysis Documentation

- [Analysis Utilities](#)
- [Validation Utility](#)

Product Tutorials

- [Configuration](#)
- [Interactive Import](#)
- [Interactive Export](#)

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

The installation is run from the .msi file download provided. For full details of the installation process, visit www.theorem.com/documentation

License Configuration

To run any product a valid license file is required. The Flex License Manager is run from the .msi file download provided. For full details of the installation process, visit www.theorem.com/documentation

Using the Product

To use the product, follow the documented steps found in this document or follow the online video tutorials which can be found from www.theorem.com/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 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 during installation.

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

<installation_directory>\bin

The script name is:

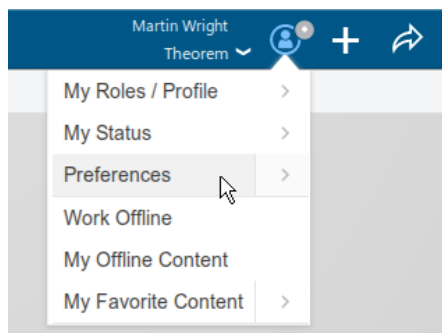
start_3DEXPERIENCE_Theorem_Multi-CAD_JT_CATIAV6R<version>.cmd

(where <version> should be substituted for the version of CATIA 3DEXPERIENCE that you have installed – e.g. 2020x, 2021x, 2022x):

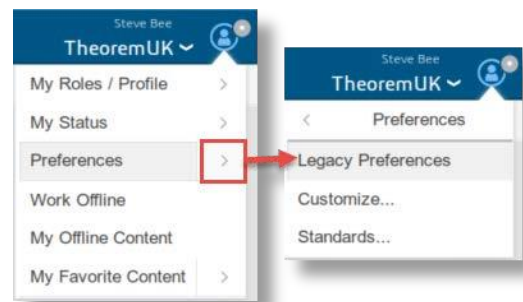
Theorem Interactive Conversion Settings

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

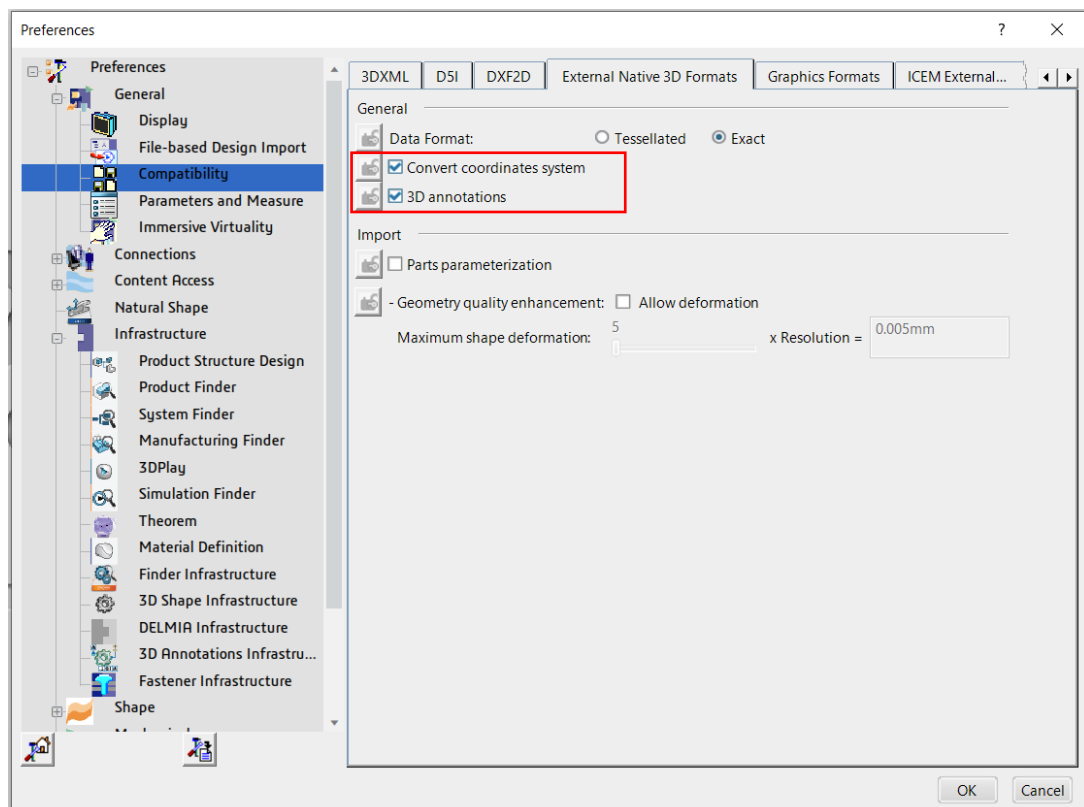
3DEXPERIENCE 2018x



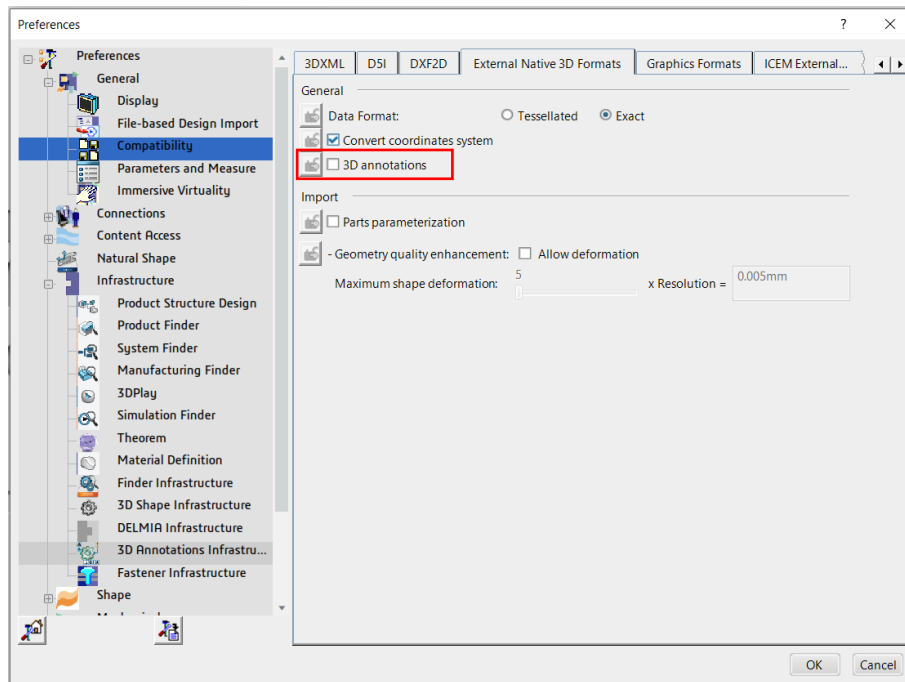
3DEXPERIENCE 2019x onwards



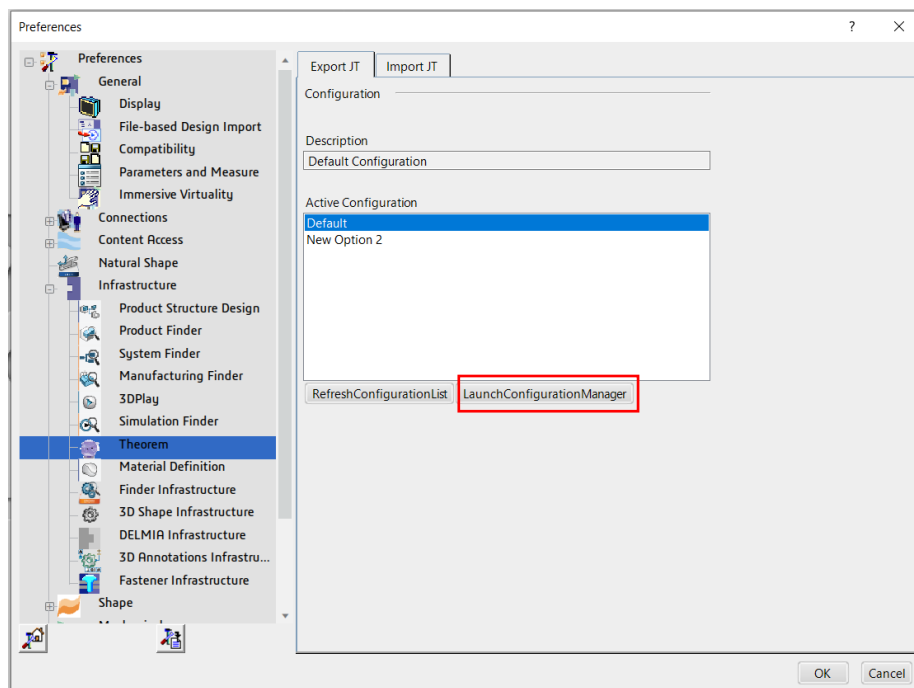
This page is a standard Dassault Page that sets the preferred mode of conversion (in this case Exact), and also the general options “Convert coordinate systems” and “3D Annotation”. Both of which are toggled on.



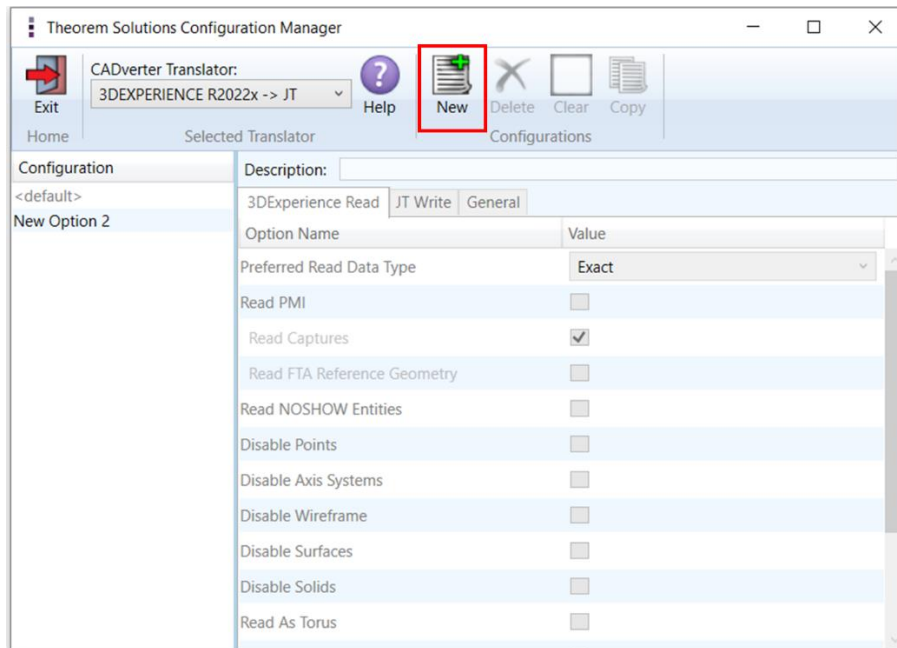
Note – Toggle off 3D Annotations to ensure that the Read PMI option is respected when using the Theorem import, i.e. PMI is displayed on the imported part when it is toggled on and is not displayed when it is toggled off.



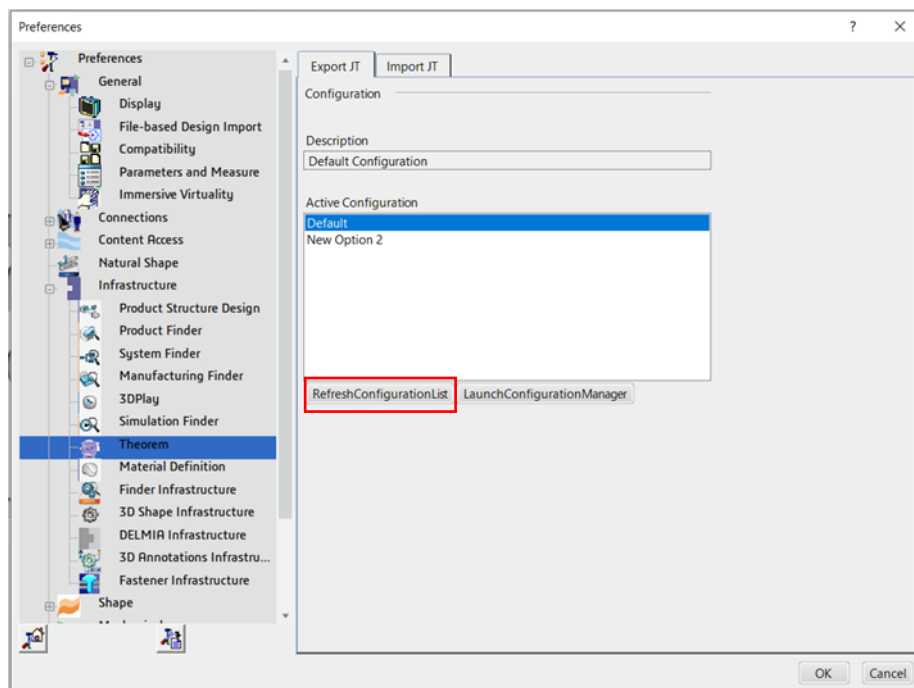
Two dedicated tabs under '**Infrastructure>Theorem**' allow the user access to Theorem Configurations for import and export. From this Panel, the user can select a predefined configuration or create a new configuration. To create a new configuration, select the '**LaunchConfigurationManager**' command.



In the configuration manager window, select New. Rename as required.



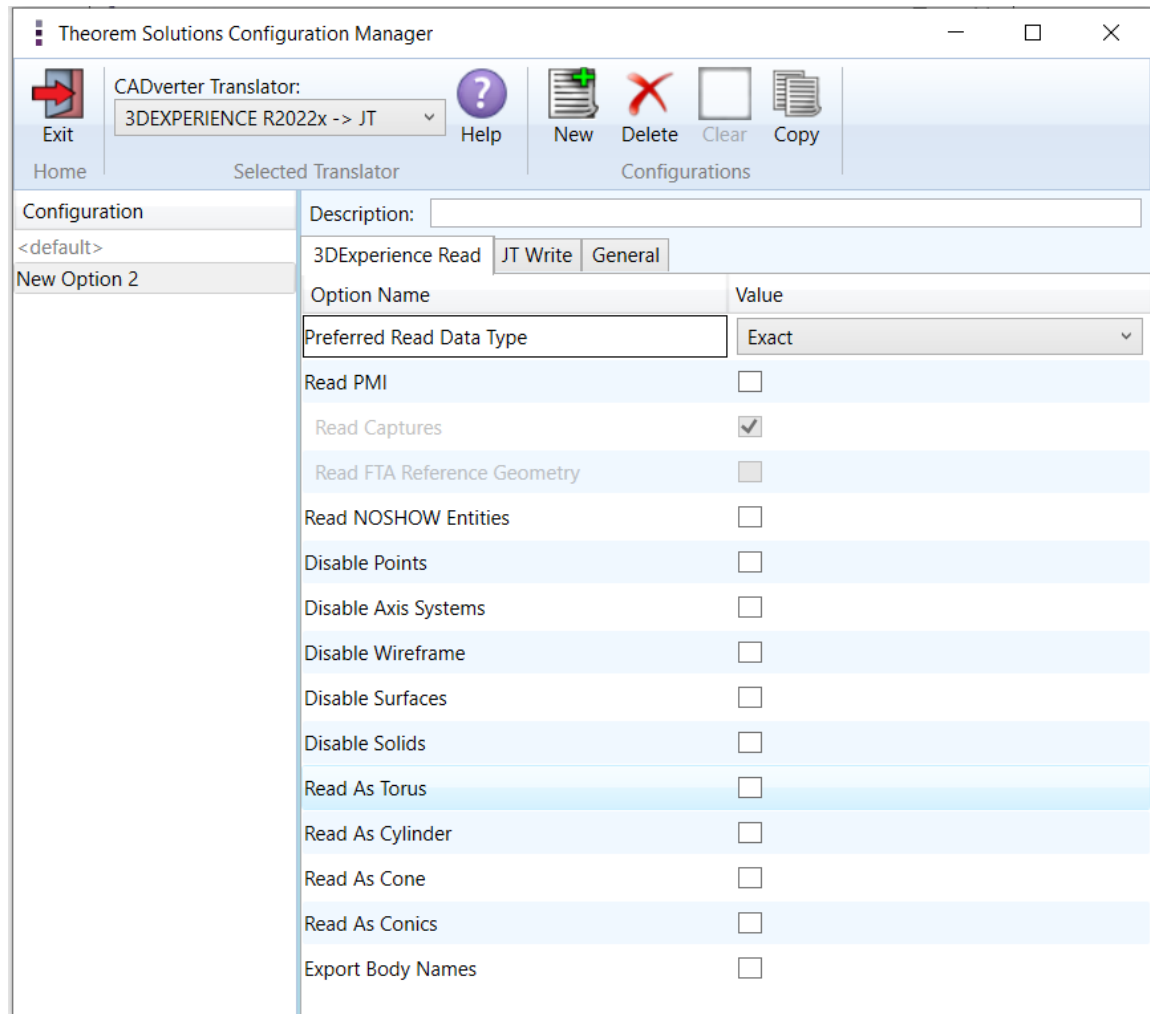
Any new configurations created will be displayed in the Active Configurations list once it has been refreshed. To do this select ***'RefreshConfigurationList'***



Configuration Manager

3DEXPERIENCE to JT Export

3DExperience Read



The screenshot shows the 'Theorem Solutions Configuration Manager' window. The 'CADverter Translator' is set to '3DEXPERIENCE R2022x -> JT'. The 'Selected Translator' is '3DExperience Read'. The 'JT Write' tab is active, showing a list of options for reading 3D data.

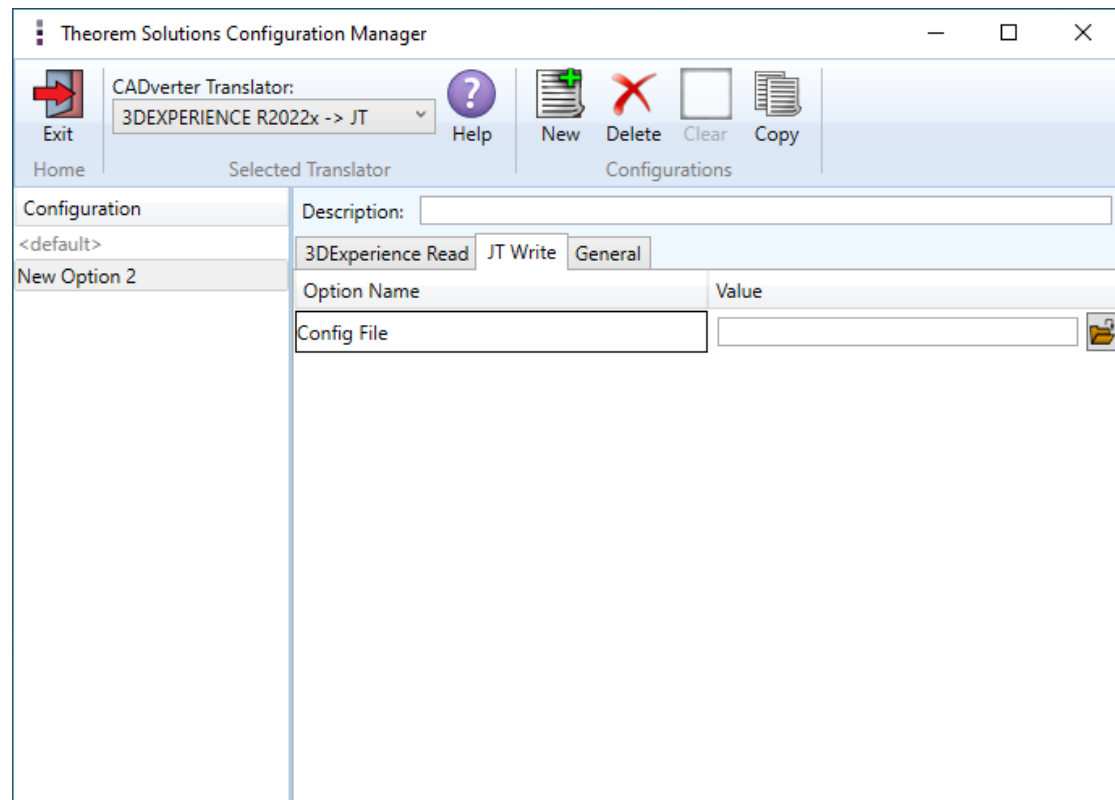
Option Name	Value
Preferred Read Data Type	Exact
Read PMI	<input type="checkbox"/>
Read Captures	<input checked="" type="checkbox"/>
Read FTA Reference Geometry	<input type="checkbox"/>
Read NOSHOW Entities	<input type="checkbox"/>
Disable Points	<input type="checkbox"/>
Disable Axis Systems	<input type="checkbox"/>
Disable Wireframe	<input type="checkbox"/>
Disable Surfaces	<input type="checkbox"/>
Disable Solids	<input type="checkbox"/>
Read As Torus	<input type="checkbox"/>
Read As Cylinder	<input type="checkbox"/>
Read As Cone	<input type="checkbox"/>
Read As Conics	<input type="checkbox"/>
Export Body Names	<input type="checkbox"/>

Each of these options is described below:

Option	Description
Preferred Read Data type	<p>The setting options are EXACT (default) or TESSELATED</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> <i>read_tess</i>
Read PMI	<p>Enables PMI data read (Default is OFF).</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> <i>dont_read_pmi</i> – to turn off <i>read_pmi</i>- to turn on <p>Note! When '<i>read_pmi</i>' is enabled it also enables the '<i>fill_pmi_arrows</i>', '<i>fill_pmi_text</i>' and '<i>pmi_filled_text</i>' options. These can be overridden by setting the Advanced arguments: '<i>dont_fill_pmi_arrows</i>' and/or '<i>dont_fill_pmi_text</i>'</p>
Read Captures	<p>A secondary argument to 'Read PMI' and allows control over whether captures are read as part of the process. Default is ON when 'Read PMI' is marked as ON. Captures can ONLY be read when 'Read PMI' is ON.</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> <i>read_captures</i> - default <i>don't_read_captures</i> – to turn off
Read FTA Reference Geometry	<p>Enables reading of FTA Reference Geometry (Default is Off). FTA Reference Geometry can ONLY be read when 'Read PMI' is ON.</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> <i>read_geometry</i> – to turn on
Read NOSHOW Entities	<p>Read any entities that are in NOSHOW. Default is to not read NOSHOW entities</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> <i>noshow</i>
Disable Points	<p>Switches off Point processing</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> <i>disable_points</i>
Disable Axis Systems	<p>Switches off Axis System processing</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> <i>disable_axes</i>

Disable Wireframe	<p>Switches off Wireframe processing</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>disable_wireframe</code>
Disable Surfaces	<p>Switches off Surface processing</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>disable_surfaces</code>
Disable Solids	<p>Switches off Solids processing</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>disable_solids</code>
Read As Torus	<p>Read Toroidal surfaces in analytical form (default is NURBS)</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>read_torus</code>
Read As Cylinder	<p>Read Cylindrical surfaces in analytical form (default is NURBS)</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>read_cylinder</code>
Read As Cone	<p>Read Cone surfaces in analytical form (default is NURBS)</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>read_cone</code>
Read As Conics	<p>Read surfaces generated from a Conic in analytical form (default is NURBS)</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>read_conics</code>
Export Body Names	<p>Maintains body names for parts that consist of multiple bodies.</p> <p>Command Line Syntax:</p> <ul style="list-style-type: none"> • <code>body_names</code>

JT Write



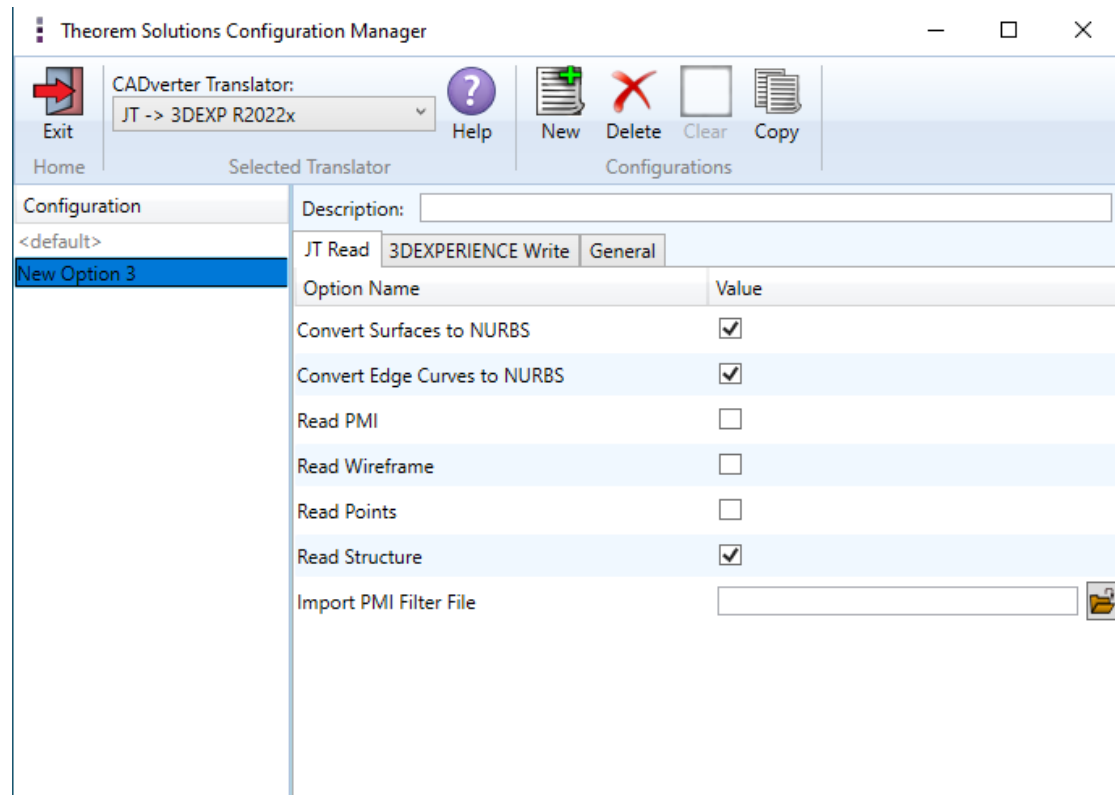
The screenshot shows the 'Theorem Solutions Configuration Manager' window. The 'CADverter Translator' dropdown is set to '3DEXPERIENCE R2022x -> JT'. The 'JT Write' tab is selected under the '3DExperience Read' category. The 'Option Name' field is 'Config File' and the 'Value' field is empty. The 'Description' field is also empty.

Option Name	Value
Config File	

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

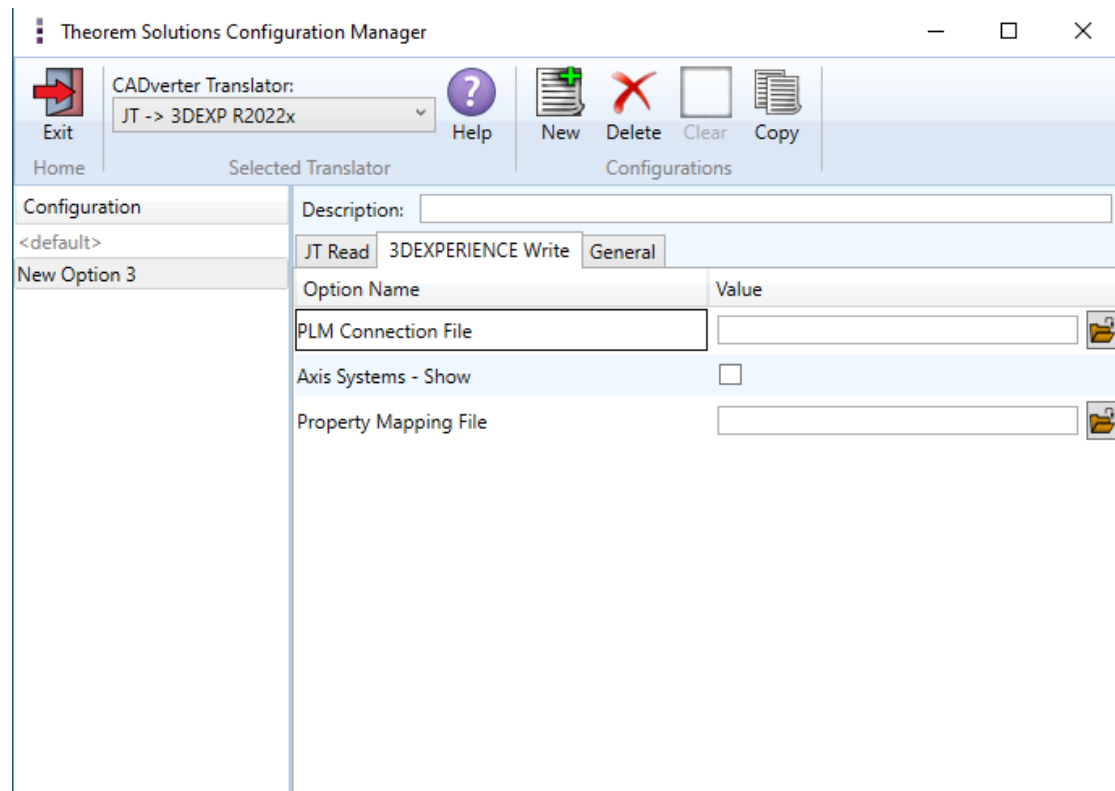
JT Read





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: <ul style="list-style-type: none"> • noprep – to turn off
Convert Edge Curves to NURBS	Read XT Brep edge curves as NURBS curves (else read in native form). Default is ON. Command Line Syntax: <ul style="list-style-type: none"> • rd_native_edge – to turn off
Read PMI	Reads 3D PMI. Default is OFF. Command Line Syntax: <ul style="list-style-type: none"> • read_pmi dim2_pmi
Read Wireframe	Read JT wireframe data. Default is OFF. Command Line Syntax: <ul style="list-style-type: none"> • read_wire_frame
Read Points	Read JT point data. Default is OFF. Command Line Syntax: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • pmi_filter_file "file name"

3DExperience Write



The screenshot shows the 'Theorem Solutions Configuration Manager' window. The 'CADverter Translator' dropdown is set to 'JT -> 3DEXP R2022x'. The 'Selected Translator' tab is active, showing a list of configurations. The 'General' sub-tab is selected, displaying a table of options.

Option Name	Value
PLM Connection File	<input type="text"/> 
Axis Systems - Show	<input type="checkbox"/>
Property Mapping File	<input type="text"/> 

Axis Systems – Show when toggled on will display the axis systems on the imported JT item in 3DExperience

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\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 REFASSTYPE attribute

PMI Type Filter

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.

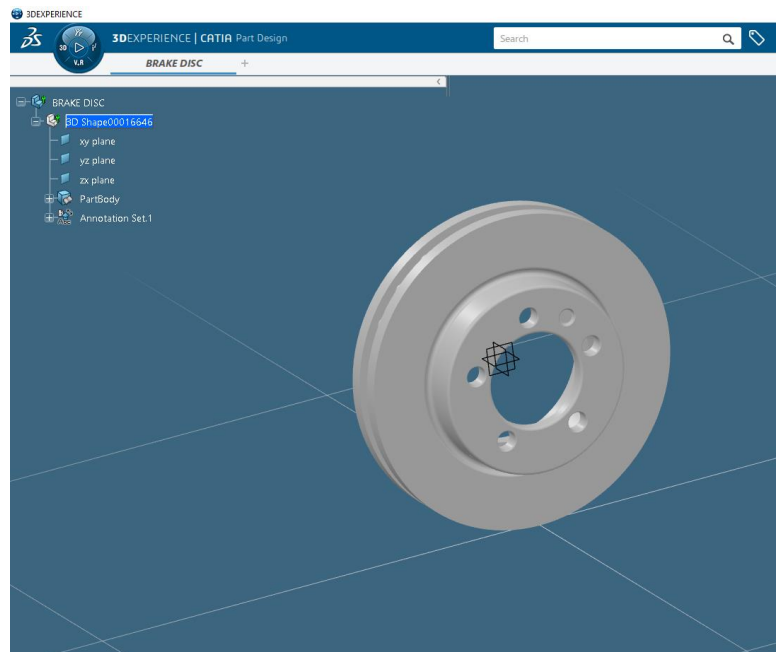
This facility was introduced to reduce processing time due to large numbers of 'redundant' PMI entities in a JT file.

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`.

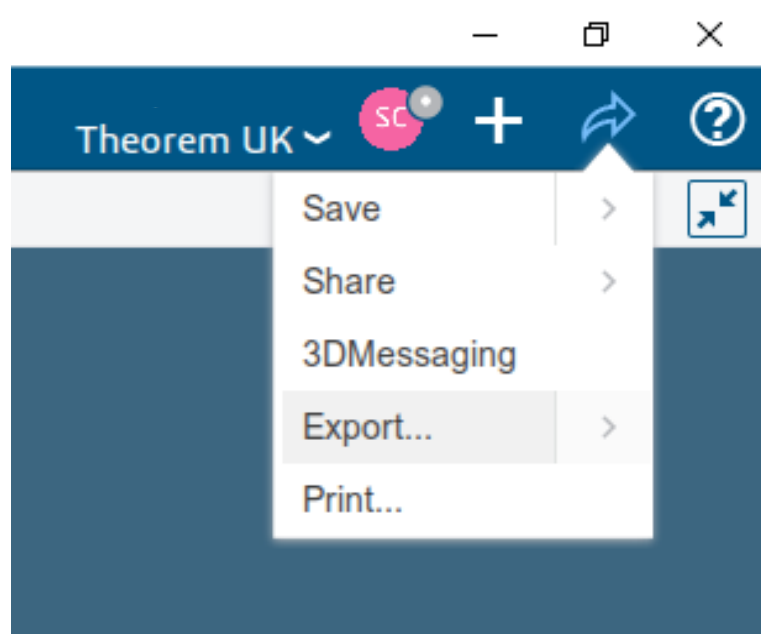
Default settings that can control the JT Import plugin, are also read from the data file `%TS_INST%\data\jt\jt_xcad_opts.txt`.

Theorem Export

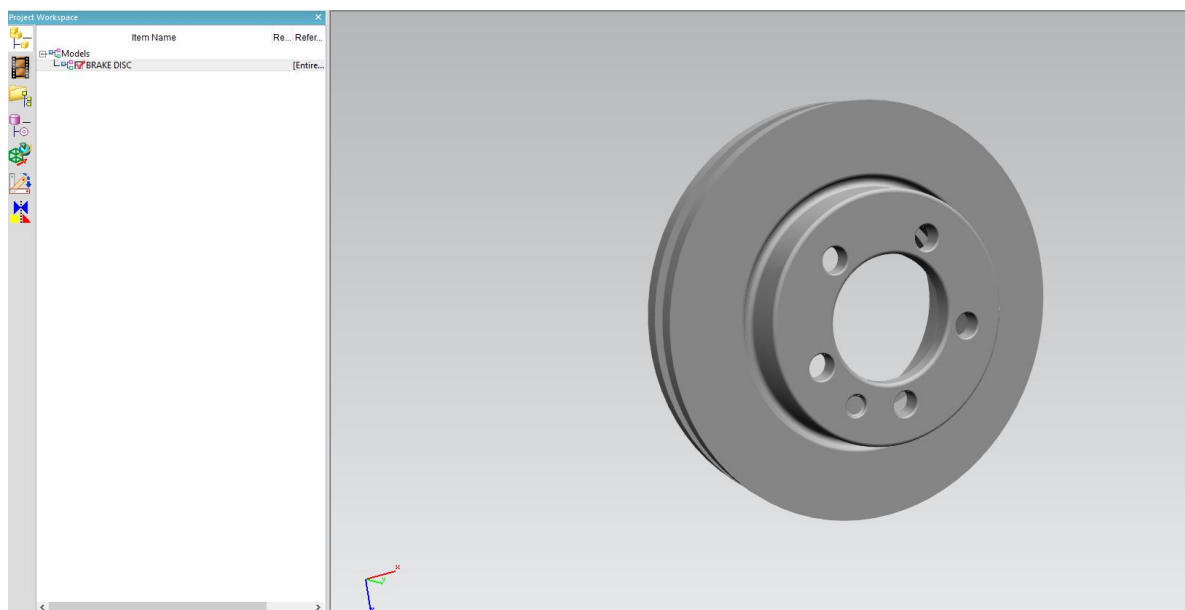
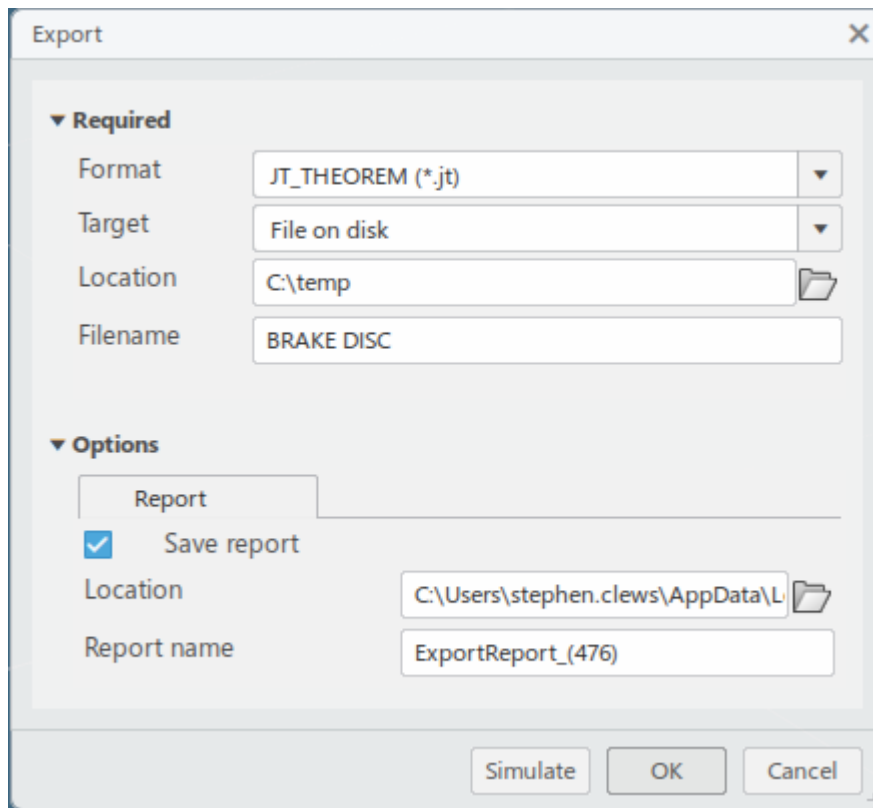
Once CATIA 3DEXPERIENCE has been started, open the product or representation that is going to be exported to JT.



Select the arrow in the top right corner, then from the list displayed select Export



Export dialog box is displayed. Ensure the “Format” is set to JT_THEOREM (*.jt). Select the required location for the file and ensure the Filename displayed is correct. Click OK to initiate the translation.



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\25.3\data\jt\template.jt`

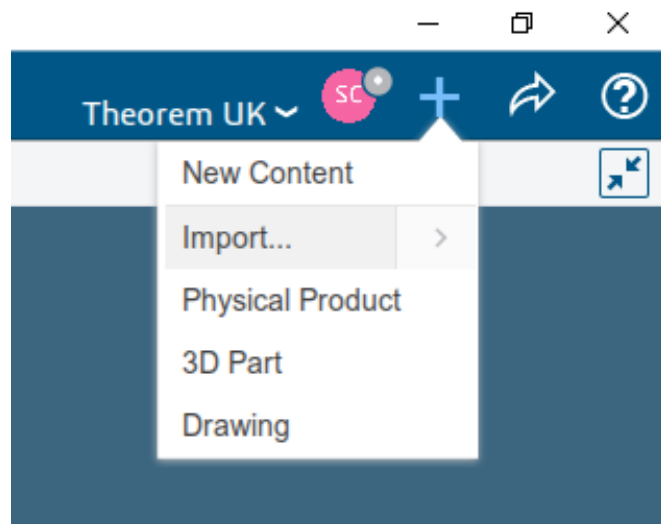
***`TS_JT_TEMPLATE_FILE_EMPTY=C:\Program
Files\Theorem\25.3\data\jt\template_empty.jt`***

***`TS_JT_TEMPLATE_FILE_HIDDEN=C:\Program
Files\Theorem\25.3\data\jt\template_hidden.jt`***

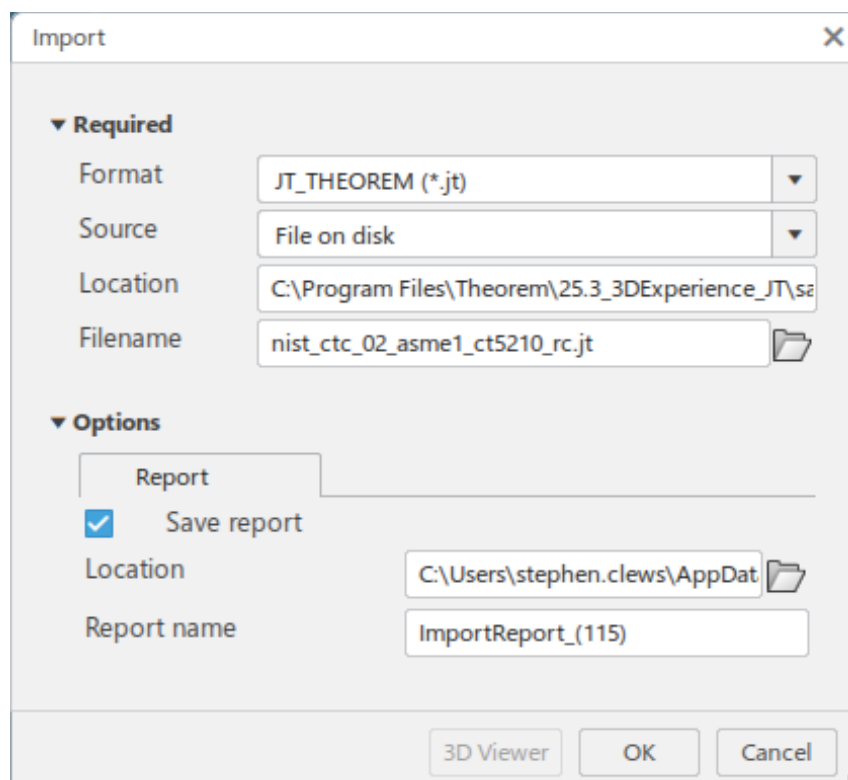
The user can change the content of these JT files or their location and names as required.

Theorem Import

Once CATIA 3DEXPERIENCE has been started it is possible for a model to be imported from JT. To do this select the '+' icon in the top right corner, then from the list displayed select import.



Import dialog box is displayed. Ensure the "Format" is set to JT_THEOREM (*.jt/.plmxml/.stpx). Click on the folder icon next to the Filename field to choose the required file for import, then click OK to initiate the translation. JT data will be imported into the CATIA 3DEXPERIENCE database and when complete will be opened into a new tab in the user's session.



Translating in Batch using CATUtil – DataExchangePLMBatch

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. 2020x, 2021x, 2022x)

Note: It can also be started from Tools>Utility within an Interactive CATIA 3DEXPERIENCE 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.

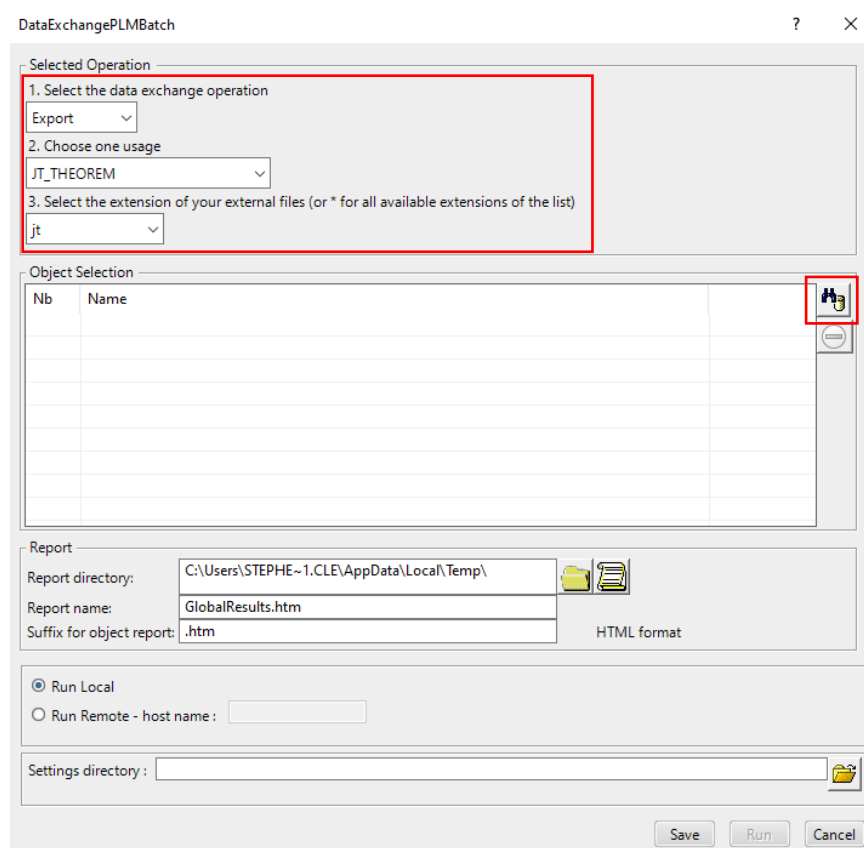
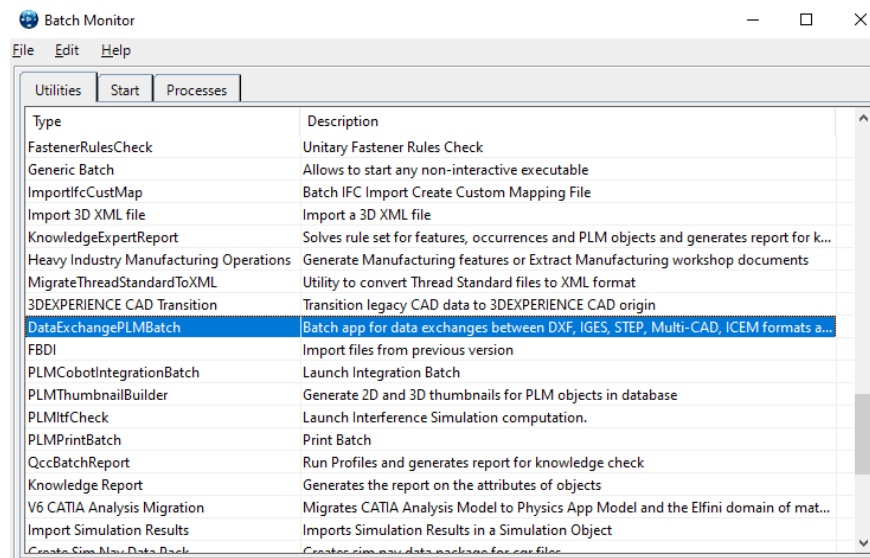
This can be achieved by setting the following variable in the CATIA Environment being used:-

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.

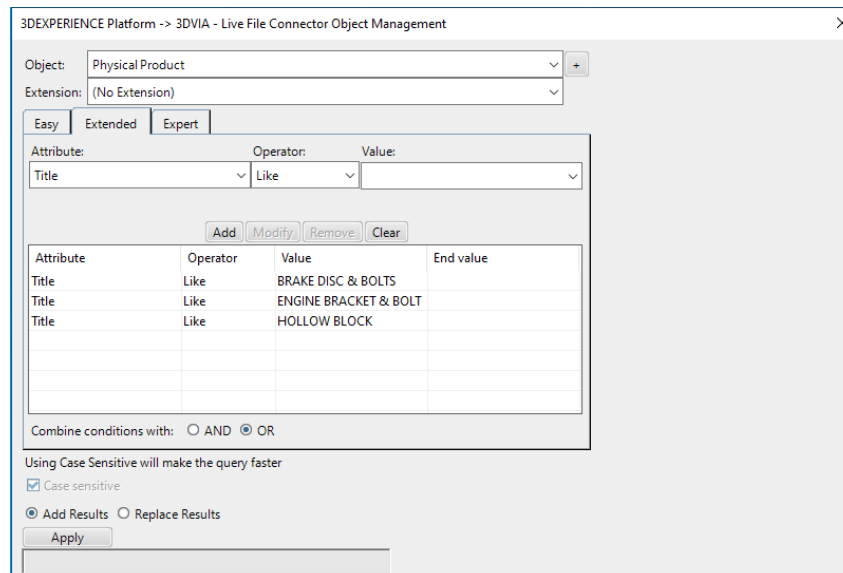
Double click on DataExchangePLMBatch to launch the command.



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

To add the required products to the batch for export, select the 3DEXPERIENCE Platform Object Search icon. 

In the subsequent window displayed, add the products using the correct attribute, operator and value ensuring that the correct object type has been selected. (E.g. Physical Product.) For multiple products ensure that the OR condition has been selected. Select Add results and click Apply to add the list of products to the Object Selection field.



3DEXPERIENCE Platform -> 3DVIA - Live File Connector Object Management

Object: Physical Product
Extension: (No Extension)

Easy | Extended | Expert

Attribute: Title Operator: Like Value:
 Add Modify Remove Clear

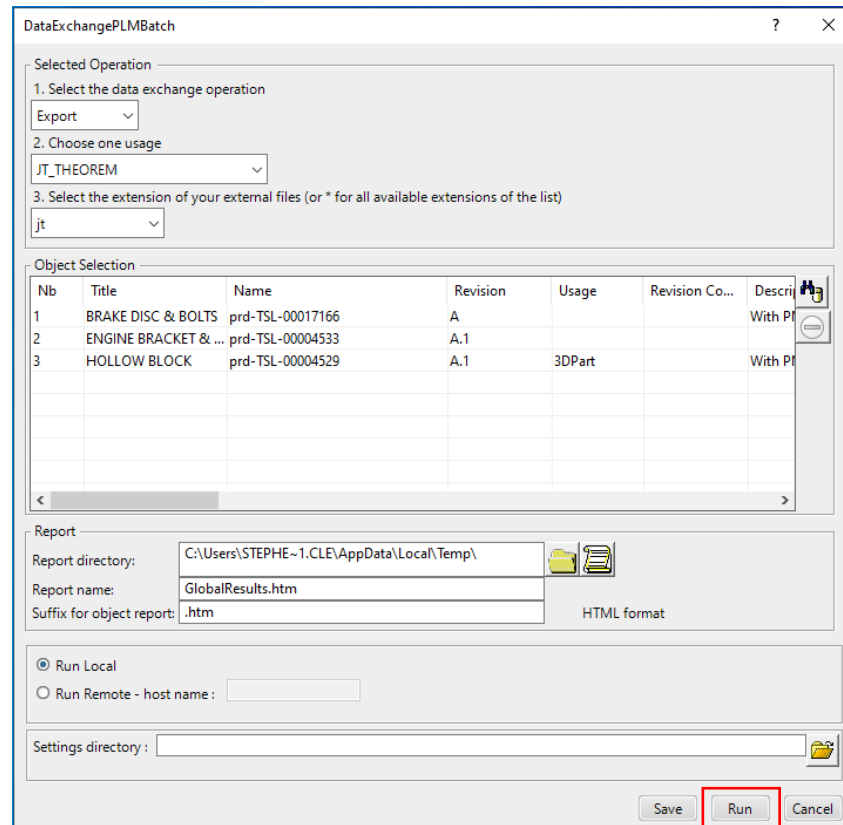
Attribute	Operator	Value	End value
Title	Like	BRAKE DISC & BOLTS	
Title	Like	ENGINE BRACKET & BOLT	
Title	Like	HOLLOW BLOCK	

Combine conditions with: ☐ AND ☒ OR

Using Case Sensitive will make the query faster
☒ Case sensitive

☒ Add Results ☐ Replace Results

Apply



DataExchangePLMBatch

Selected Operation

1. Select the data exchange operation
Export


2. Choose one usage
JT_THEOREM

3. Select the extension of your external files (or * for all available extensions of the list)
jt

Object Selection

Nb	Title	Name	Revision	Usage	Revision Co...	Descri
1	BRAKE DISC & BOLTS	prd-TSL-00017166	A			With P
2	ENGINE BRACKET & ...	prd-TSL-00004533	A.1			With P
3	HOLLOW BLOCK	prd-TSL-00004529	A.1	3DPart		With P


Report

Report directory: C:\Users\STEPHE~1.CLE\AppData\Local\Temp\ 

Report name: GlobalResults.htm

Suffix for object report: .htm HTML format

☒ Run Local
☐ Run Remote - host name:

Settings directory: 

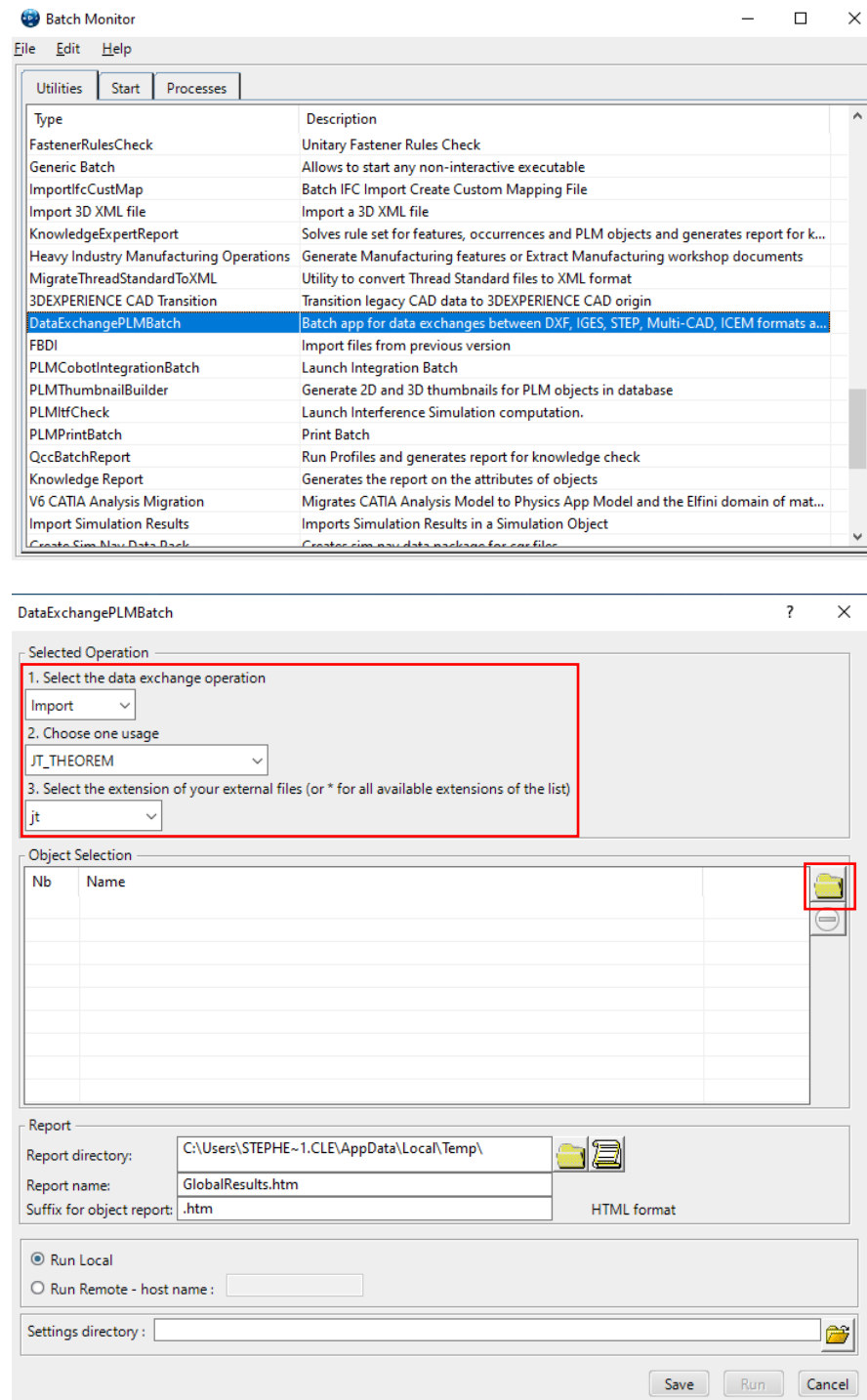
Save Run Cancel

Click Run to initiate the Batch Export to JT.


Batch Import from JT

Here is an example showing the appropriate settings in preparation for an input JT file selection and an import.

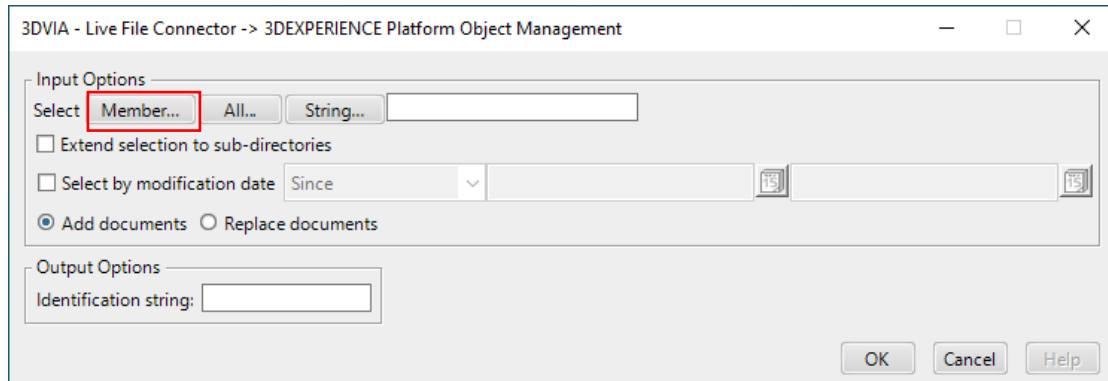
Double click on DataExchangePLMBatch to launch the command.



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

To add the required products to the batch for import, select the 3DVIA – Live File Connector Object Search icon, next to the Object Selection field. 

In the subsequent window displayed, add the products by clicking on the Member and selecting the required JT files. These will then be added to the Object Selection field.



3DVIA - Live File Connector -> 3DEXPERIENCE Platform Object Management

Input Options

Select **Member...** All... String...

☐ Extend selection to sub-directories

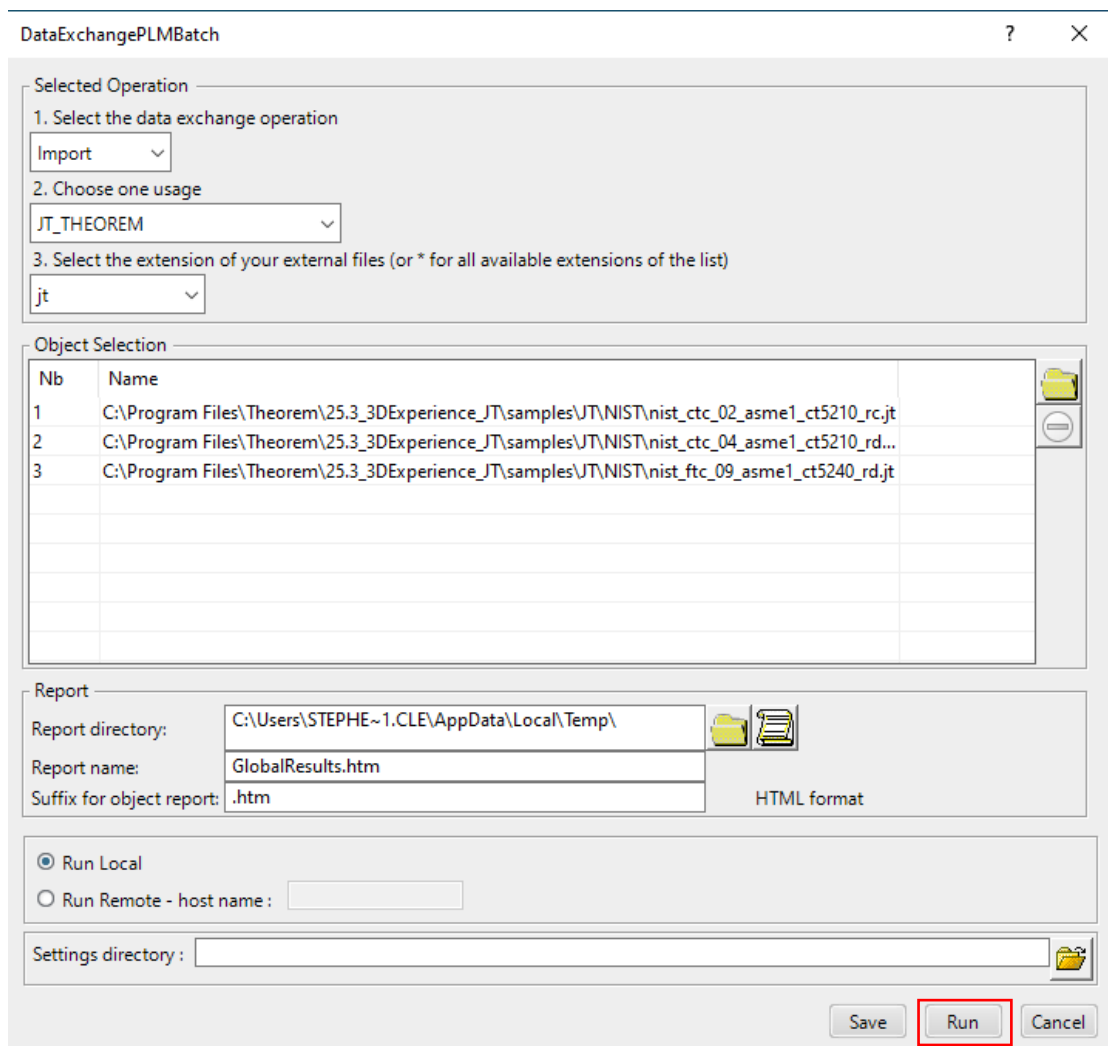
☐ Select by modification date Since [dropdown] [calendar icon]

☒ Add documents ☐ Replace documents

Output Options

Identification string: [text box]

OK Cancel Help



DataExchangePLMBatch

Selected Operation

1. Select the data exchange operation
Import

2. Choose one usage
JT_THEOREM

3. Select the extension of your external files (or * for all available extensions of the list)
jt

Object Selection

Nb	Name
1	C:\Program Files\Theorem\25.3_3DExperience_JT\samples\JT\NIST\nist_ctc_02_asme1_ct5210_rc.jt
2	C:\Program Files\Theorem\25.3_3DExperience_JT\samples\JT\NIST\nist_ctc_04_asme1_ct5210_rd...
3	C:\Program Files\Theorem\25.3_3DExperience_JT\samples\JT\NIST\nist_ftc_09_asme1_ct5240_rd.jt

Report

Report directory: C:\Users\STEPHE~1.CLE\AppData\Local\Temp\

Report name: GlobalResults.htm

Suffix for object report: .htm HTML format

☒ Run Local
☐ Run Remote - host name: [text box]

Settings directory: [text box]

Save **Run** Cancel

Click Run to initiate the Batch Import from JT.

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.

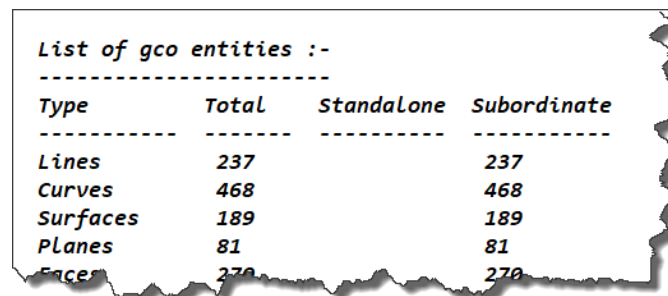
e.g. %CATReport%\model1.err

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 are 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>\B422\win_b64\CATEnv\Theorem_Multi-CAD_JT_CATIAV6R2020x.txt

<installation_directory>\B423\win_b64\CATEnv\Theorem_Multi-CAD_JT_CATIAV6R2021x.txt

<installation_directory>\B424\win_b64\CATEnv\Theorem_Multi-CAD_JT_CATIAV6R2022x.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_v6.txt` file (where `TS_INST` = the Theorem Installation folder).

By default, this is set to `TS_INST\etc\tess.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\tess.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" "ABSOLUTE"	chordalOption = "RELATIVE"
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	true TRUE false FALSE	seamSewing = true
Note: Not available for Unigraphics.		
seamSewingTol	<i>any integer</i>	seamSewingTol = 0.001
includeBrep	true TRUE false FALSE	includeBrep = false
brepPrecision	"SINGLE" "DOUBLE"	brepPrecision = "SINGLE"
autoNameSanitize	true TRUE false FALSE	autoNameSanitize = true
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	<i>any integer</i>	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 " <i>lod number</i> " {	LOD "1" {
Level	<i>any integer</i>	Level = 1
Chordal	<i>any number</i>	Chordal = 0.001
Angular	<i>any number</i>	Angular = 25
Length	<i>any number</i>	Length = 1
FeatureSuppression	<i>any integer</i>	FeatureSuppression = 0
Simplify	<i>any number</i>	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	<i>"string of characters"</i>	FilenameSanitizeSet =
FilenameSanitizeSetAdd	<i>"string of characters"</i>	FilenameSanitizeSetAdd = "41"
FilenameSanitizeSetDelete	<i>"string of characters"</i>	FilenameSanitizeSetDelete = "c"
MetadataKey	<i>"string of characters"</i>	MetadataKey = "metadata key to exclude"
close brace		

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	<i>"string of characters"</i>	AddToParts = "<metadata key>" AddToParts = "<metadata value>"
AddToAssemblies	<i>"string of characters"</i>	AddToAssemblies = "<metadata key>" AddToAssemblies = "<metadata value>"
AddToAllNodes	<i>"string of characters"</i>	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 only the assembly structure 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 “**xcad_jt_opts.txt**” file.

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)	<p>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.</p> <p>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.</p>
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\25.3\B424\win_b64\CATEnvTheorem_Multi-CAD_JT_CATIAV6R2022x.txt

to suit your installation

TS_V6_LAP_INPUT_TEMPLATE=C:\PPC\Products\MC6JT\master_template.xml

TS_PLM_PRODUCT_REF_TYPE_NAME= VPMReference

TS_PLM_REPRESENTATION_TYPE_NAME= VPMRepReference

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="REeYnZM3M0Q1REM2NDgxQzIFNzk1QzlwNjZGRDYz0DN8Um9iaW58Um9iaW58fHw
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"/>
    <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\catia6r<revision>_jt.cmd <input_file> -o <output_file> -z <config_file> <options>
```

Where: - <TS_INST> is the Theorem Solutions software installation directory.

Where: - <revision> is the version of 3DEXPERIENCE that you have installed – e.g. 2020x, 2021x, 2022x

<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 (2015x onwards), 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

```
<?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="REeYNzM3M0Q1REM2NDgxQzlfNzk1QzlwNjZGRDYzODN8Um9iaW58Um9ia
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"/>
    <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. -z <config_file> not required in the command line if using the default configuration.

<options>

Options displayed in the configuration manager can be added at the end of the command using the correct syntax, e.g. read_pmi, read_captures

progress_file <file name>

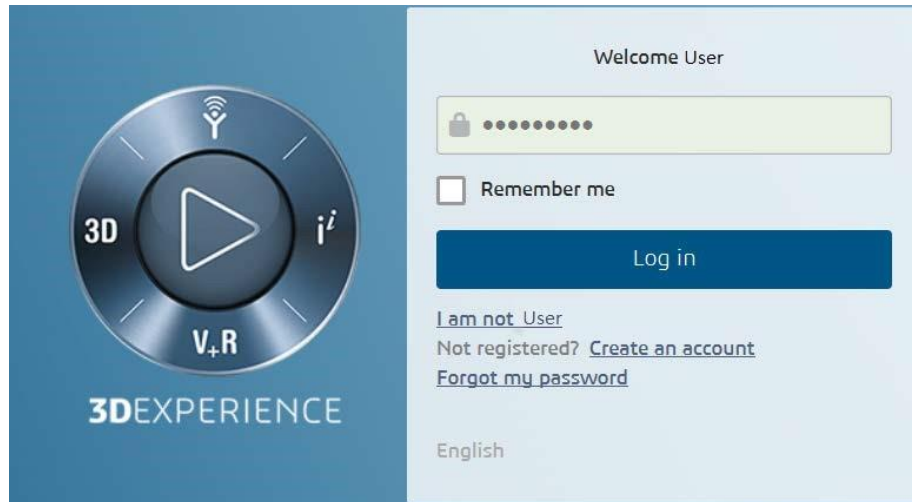
The path and file name for the log file e.g. C:\TEMP\progress.log

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

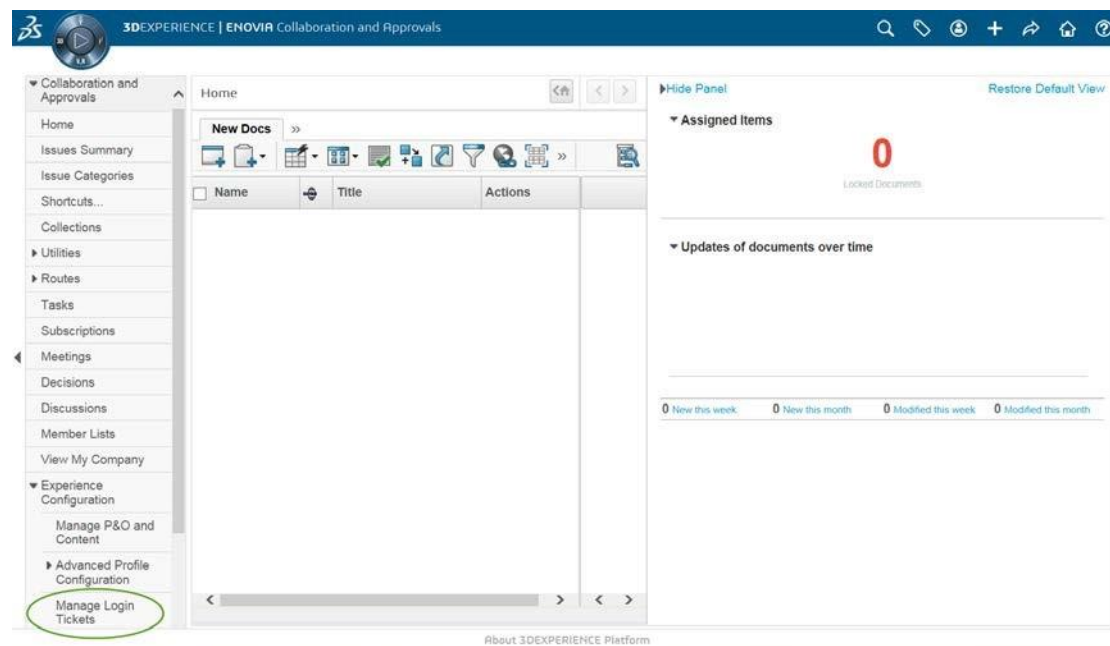



Login as normal, e.g.

User

Password

Select Collaboration and Approvals > Experience Configuration > Manage Login Tickets




3DEXPERIENCE | ENOVIA Collaboration and Approvals

Collaboration and Approvals

Home

Issues Summary

Issue Categories

Shortcuts...

Collections

Utilities

Login Ticket Creation

User:	<input type="text" value="User"/>
Security Context:	<input type="text" value="VPLMProjectLeader.Company Name.Acme"/> ▼
Ticket type:	<input checked="" type="radio"/> Infinite <input type="radio"/> Once
<input type="button" value="Create"/>	

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:-

REEyNzM3STE1MER2NDgxQzIFNzk1QzlwNjZGNATzDAN8Um9iaW58Um9iaW58fHwwfA==

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