SPINFIRE CONVERT

Powered by Theorem Technology

User Guide

3DExperience - JT

Product Category	Spinfire Convert
Product Group	3DExperience <> JT
Product Release Version	28.2

Document Type	User Guide
Document Status	Released
Document Revision	1.0
Document Author	Product Manager
Document Issued	19/01/2026

THEOREM HOUSE
MARSTON PARK
BONEHILL RD
TAMWORTH
B78 3HU
UNITED KINGDOM

+44(0)1827 305 350

THEOREM SOLUTIONS INC.
100 WEST BIG BEAVER
TROY
MICHIGAN
48084
USA

+(513) 576 1100

Contents

SpinFire's Product Suite

SpinFire have 5 main Product brands. These are:

SPINFIRE PUBLISH

Automatically Process and Publish Inbound CAD files (Formerly CAD Publisher)

SPINFIRE CONVERT

Seamless data exchange between CAD and Visualization (Formerly Theorem CADTranslate & CADPublish)

SPINFIRE MANAGE

Centralised and unified view of product data (Formerly Centro)

SPINFIRE INSIGHT

Comprehensive interrogation of 2D and 3D CAD Files (Formerly Spinfire Ultimate)

SPINFIRE EXTENDED REALITY - XR

Augmented, Mixed and Virtual Reality for Engineering (Formerly Theorem XR & Theorem TVP)

See our website for more details on these products.

About SpinFire Convert



SpinFire Convert 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.

SpinFire Convert 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.

3DExperience to JT – SpinFire Convert

The 3DExperience to JT Bi-directional Translator

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

The 3DExperience to JT Translator is a bi-directional direct database converter between Dassault Systemes 3DExperience application and 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 3DExperience data into the JT format.

The translator can be invoked either interactively or in 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 such as meta-data, colour and layer information and 3DExperience properties.
- Data can be filtered by layer and entity type during processing. Geometry can be filtered and selectively processed.
- Integrated with the 3DExperience installation.
- The conversion process can be run Interactively or in Batch Mode.
- Uses the 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 all related business processes.

This document will focus on the 3DExperience to JT bi-directional product. For further information on other SpinFire Convert products please contact sales.applications@techsoft3d.com

Getting Started

Documentation & Installation Media

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

The latest copy of the User Guide for this particular product can be found on our web site at:

https://docs.techsoft3d.com/spinfire/convert/catia_3dx/jt/user_guide.html

Each product has a specific link to the Product Release Notes, which in turn contains a link to the download location of the software.

The latest version of the 3DExperience to JT software can be found via the link below:

https://docs.techsoft3d.com/spinfire/convert/catia_3dx/jt/product_release_notes.html

Installation

The installation is run from the .msi file download provided. For full details of the installation process, visit https://docs.techsoft3d.com/spinfire/convert/catia_3dx/jt/index.html

License Configuration

To run any product a valid license file and a Flex License Manager installation will be required. The Flex License Manager is run from the .msi file download provided. This can be accessed from the Product Release Notes. For full details of the installation process, visit https://docs.techsoft3d.com/spinfire/convert/catia_3dx/jt/index.html

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 https://docs.techsoft3d.com/spinfire/convert/catia_3dx/jt/index.html

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 3DExperience**
  - The Interactive Interface provides a direct method of importing to and exporting from 3DExperience.

- **In Batch via CATUTIL - DataExchangePLMBatch**
  - The 3DExperience DataExchangePLMBatch Interface provides a direct method of invoking the translator. It can be used to translate single or multiple objects at once.

- **On the Command Line**
  - A command line method of Invoking the translator is possible.

Translating Interactively from within 3DExperience

Launching 3DExperience with SpinFire Convert Plug-ins

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

In order to translate from within 3DExperience, the application must be started using a SpinFire Convert environment, so that the appropriate plug-ins are available. ***(See 3DExperience Environment files.)***

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

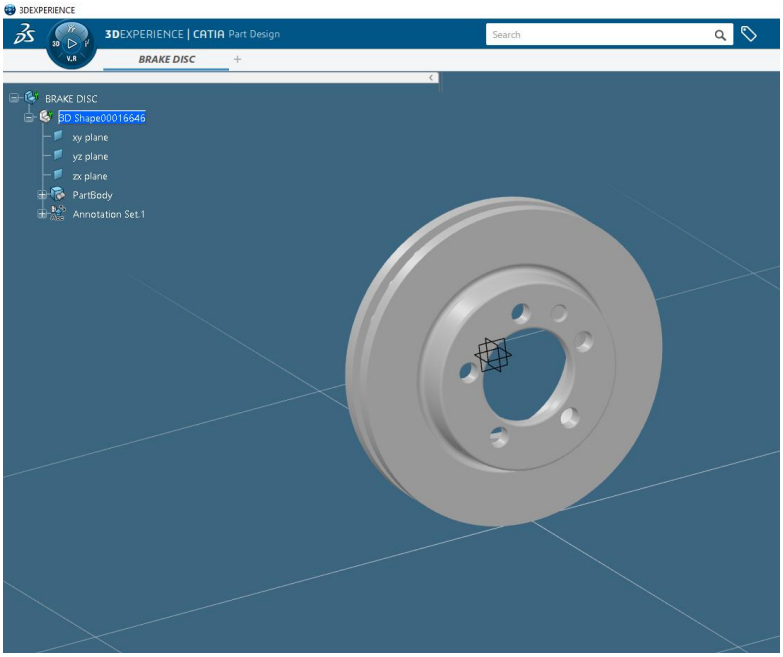
The script name is:

start_3DEXPERIENCE_<version>_JT.cmd

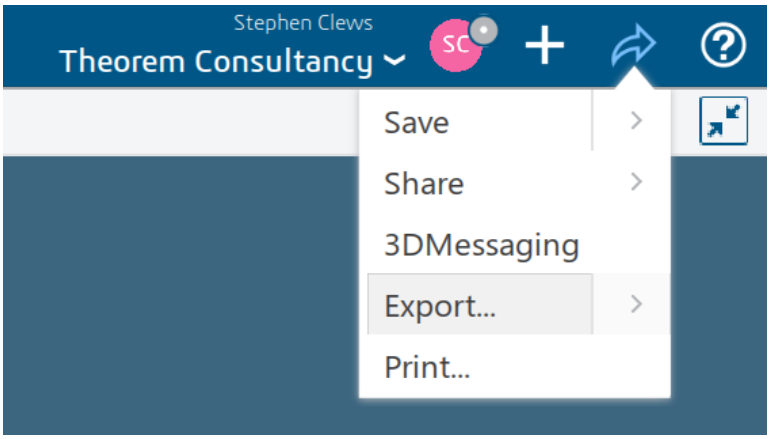
Where *<version>* is the version of 3DExperience that you have installed – e.g. 2023x, 2024x, 2025x.

Interactive Export to JT

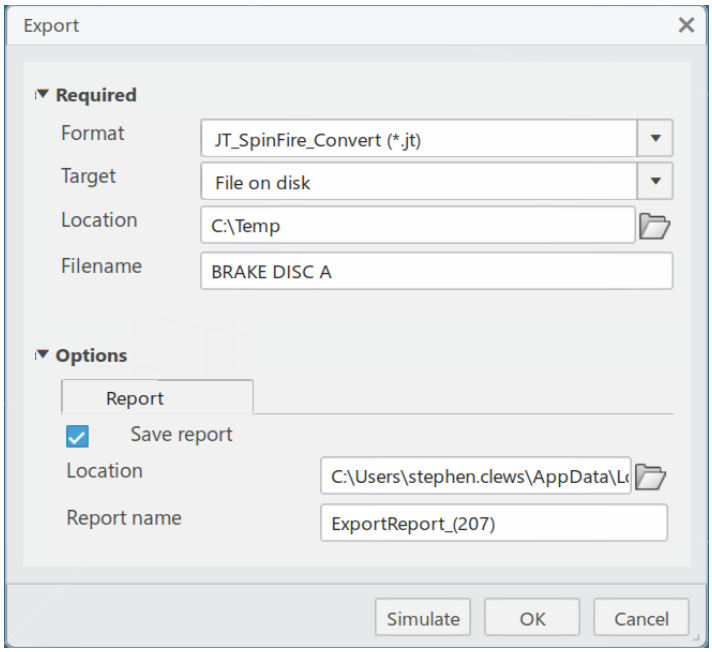
Once the 3DExperience application has been launched, open the product or representation that is going to be exported to JT.



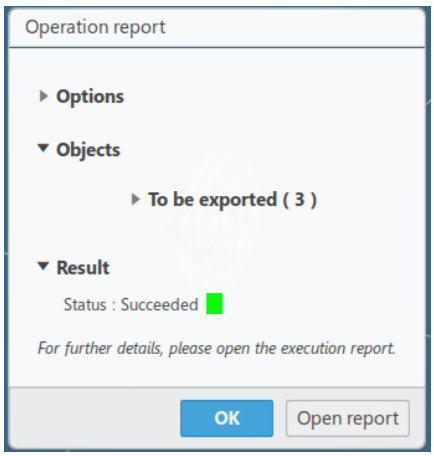
Select the arrow in the top right corner, then from the list displayed select Export.

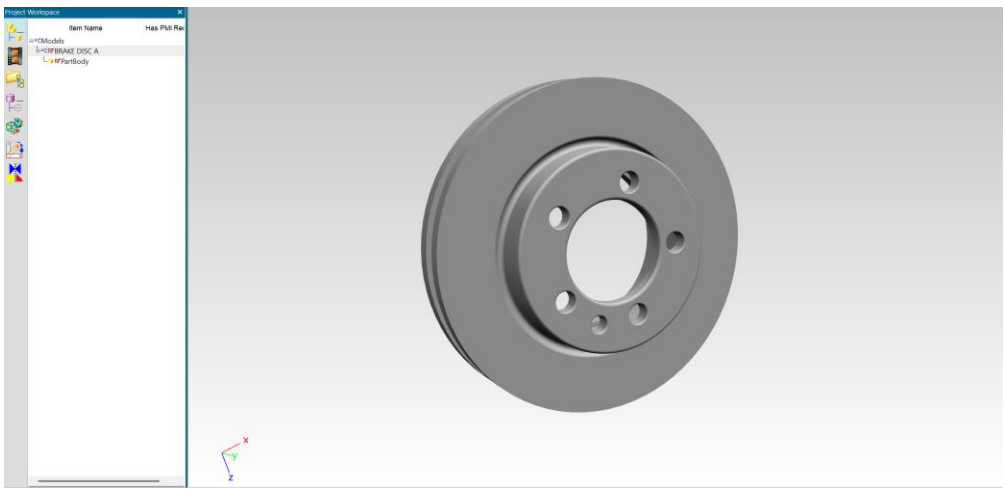


In the Export dialog box displayed, ensure the 'Format' is set to JT_SpinFire_Convert (*.jt/*.plmxml/*.stpx). Select the required location for the file and ensure the Filename displayed is correct. Click OK to initiate the export to JT.



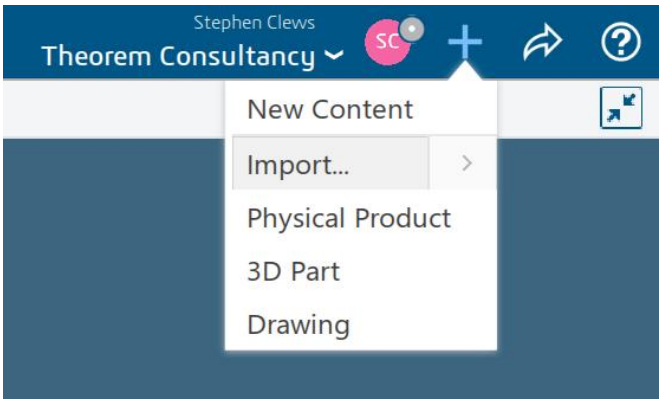
Export status of Succeeded displayed in Operation report window. JT files created in the location specified.



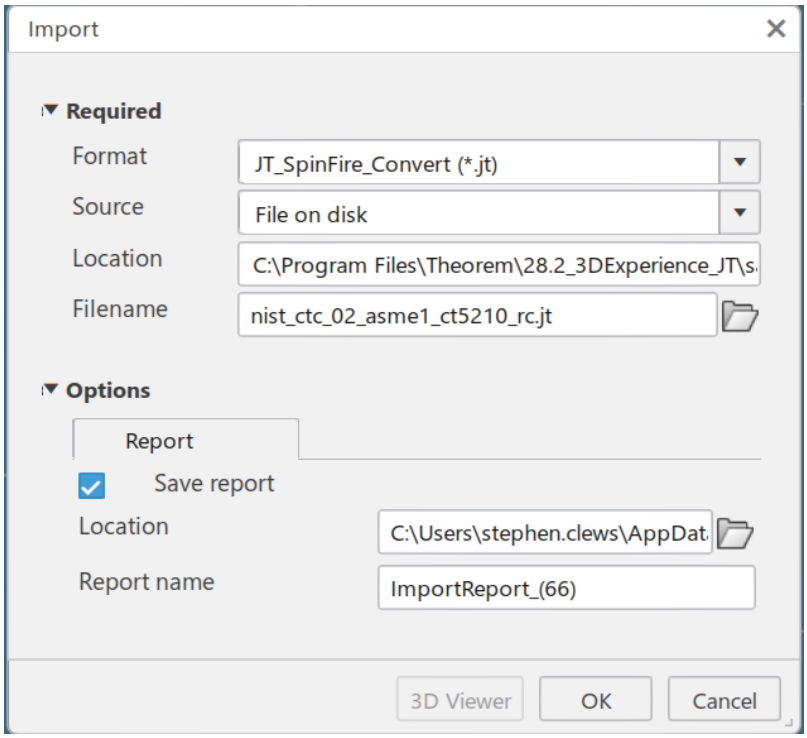


Interactive Import to 3DExperience

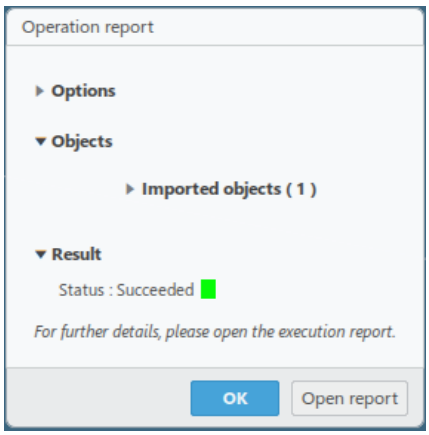
Once the 3DExperience application has been launched, it is possible for a model to be imported from JT. To do this select the **'+'** icon in the top right corner, then from the list displayed select import.

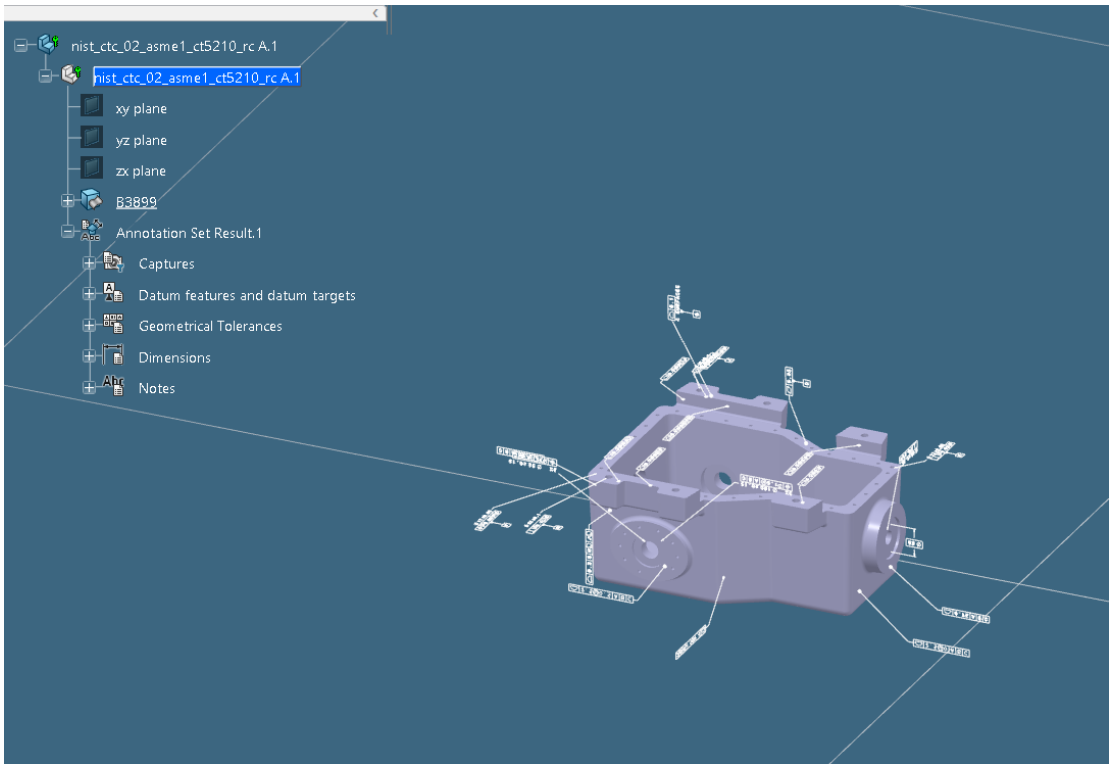


In the Import dialog box displayed, ensure the 'Format' is set to JT_SpinFire_Convert (*.jt/*.plmxml/*.stpx). Click on the folder icon next to the Filename field and select the required JT file, then click OK to initiate the import to 3DExperience.



Import status of Succeeded displayed in Operation report window. Imported data saved into the 3DExperience database and opened into a new tab in the user's session.



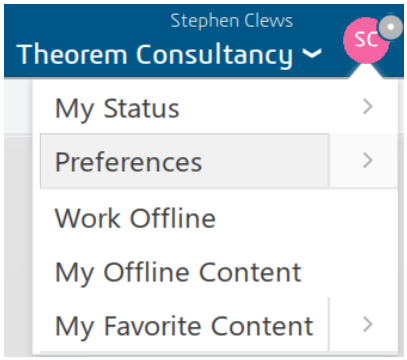


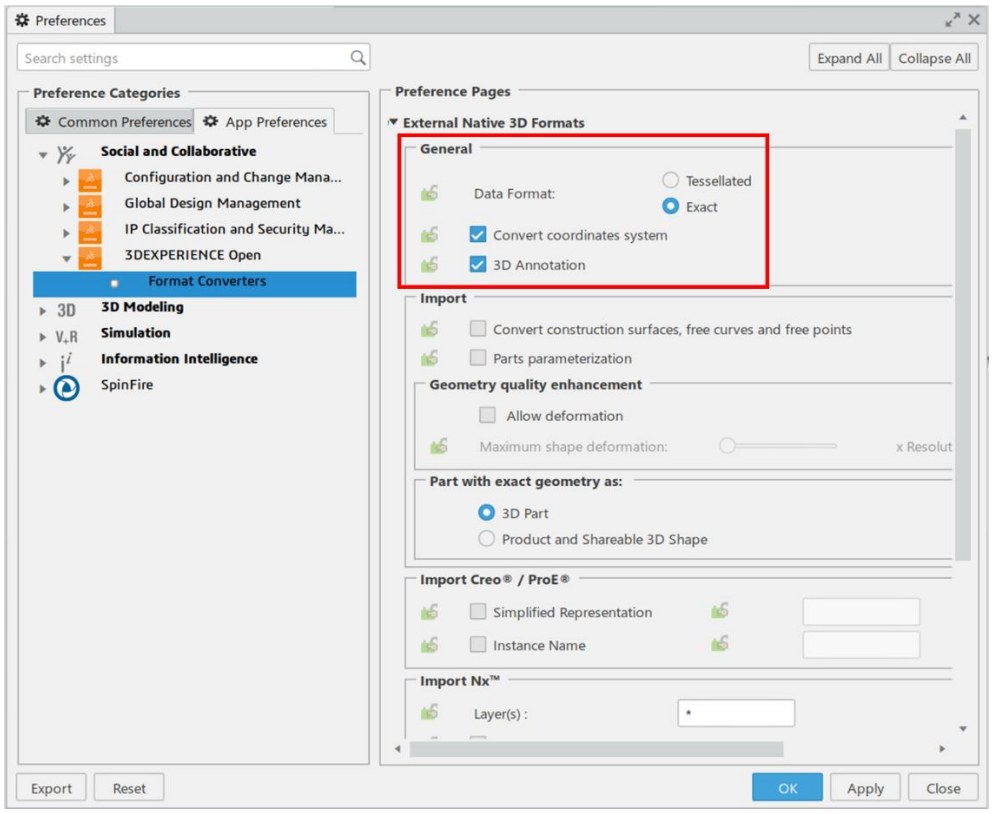
Please note that from version 28.2 of the 3DExperience to JT translator you will need to use the JT_SpinFire_Convert format in order to export data from 3DExperience and import data into 3DExperience successfully. Using the JT_THEOREM format which may still be displayed in the list will produce errors and ultimately an unsuccessful export or import.

For earlier versions of the software, the JT_THEOREM format will still need to be used. Please see **Appendix G** for more information on this.

SpinFire Convert 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>App Preferences>Social and Collaborative>3DEXPERIENCE Open>Format Converters>External Native 3D Formats.***



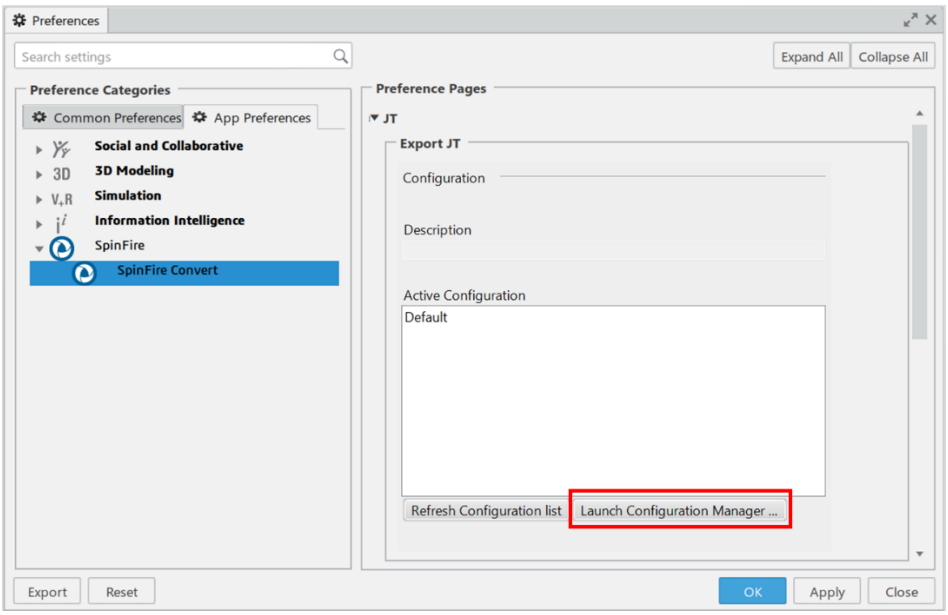


This page is a standard Dassault Page that enables the user to set the preferred mode of conversion (in this case Exact). It also enables the user to apply general options such as ***'Convert coordinates system'*** and ***'3D Annotation'***.

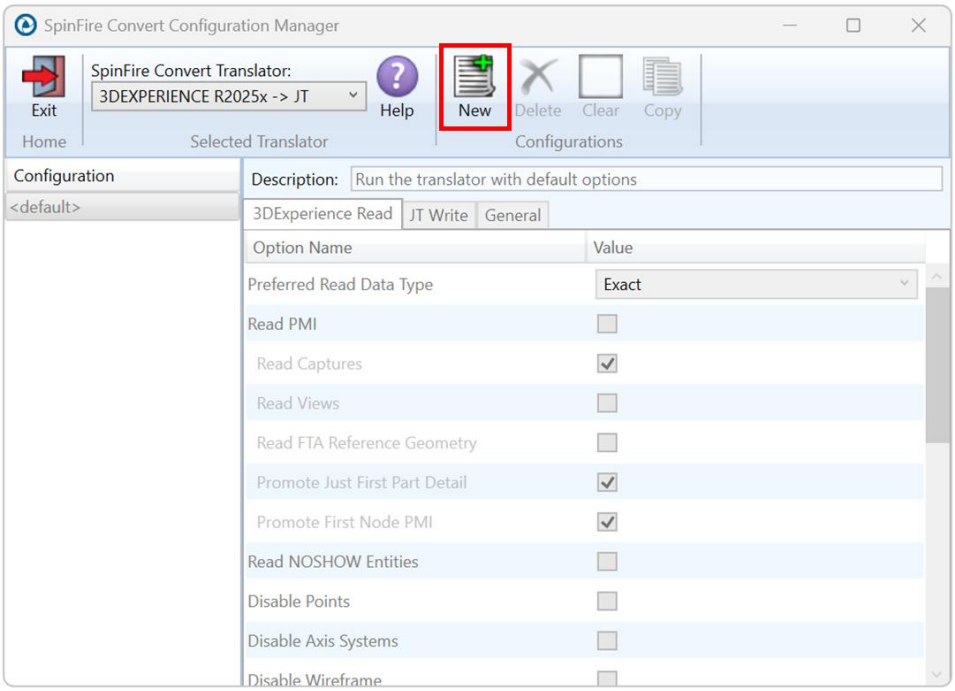
There are also some SpinFire Convert settings that can be applied. These are accessed through ***Preferences>App Preferences>SpinFire>SpinFire Convert.***

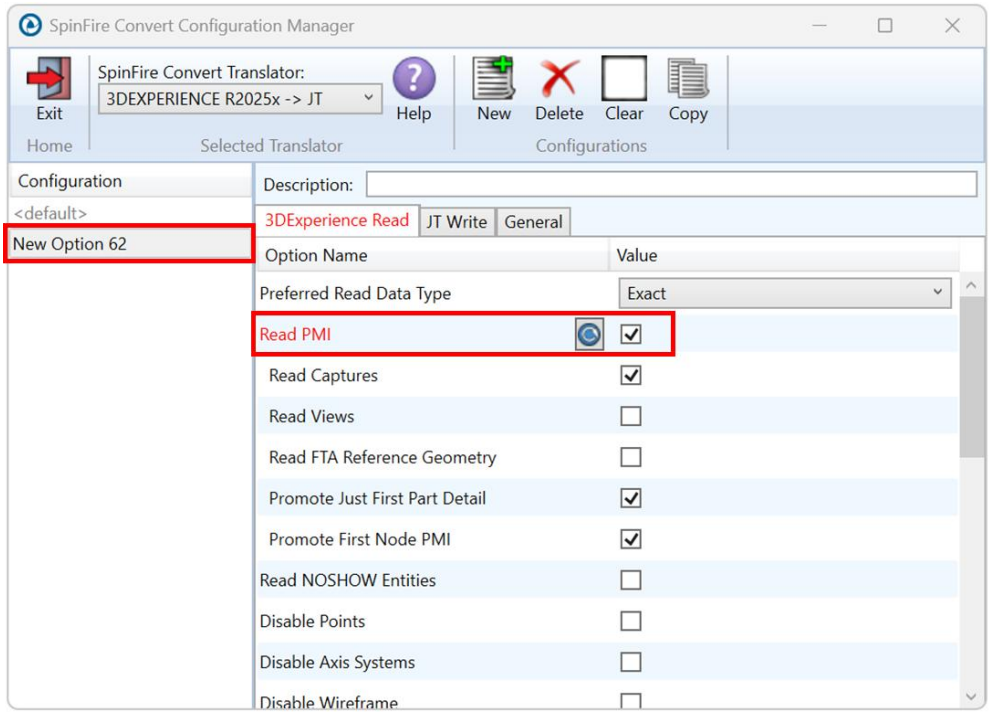
A JT drop down menu under ***'SpinFire Convert'*** allows the user access to any active configurations for both JT Export and JT Import with 2 separate active configuration windows displayed for both options. From this menu, the user can select a predefined configuration or create a new configuration for exporting to JT and importing from JT.

To create a new configuration, select the ***'Launch Configuration Manager'*** command.



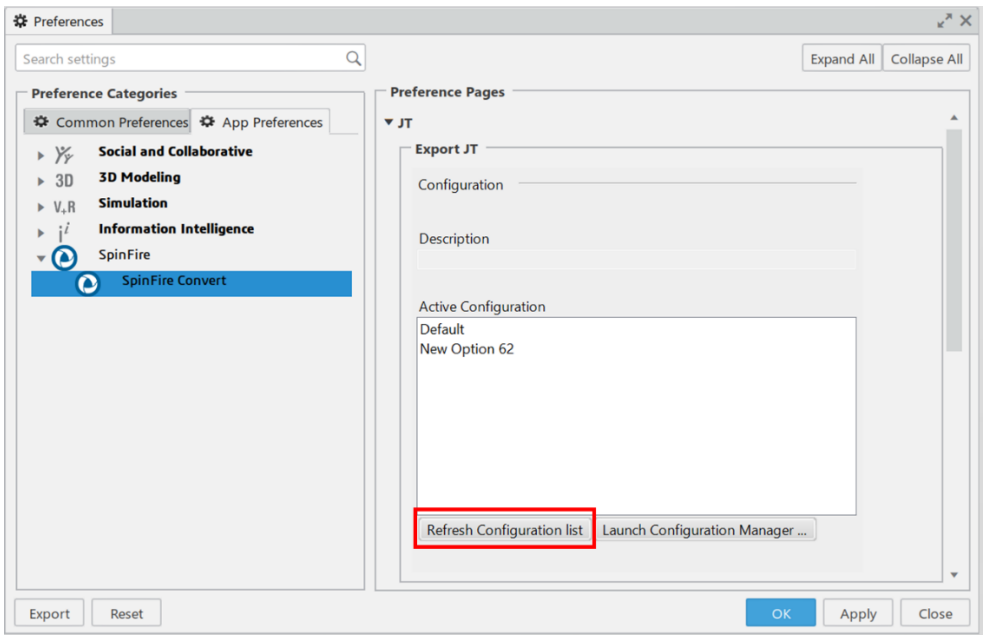
In the configuration manager window, select New, apply the relevant options, ***(see Configuration Manager)***, then rename the configuration as required.



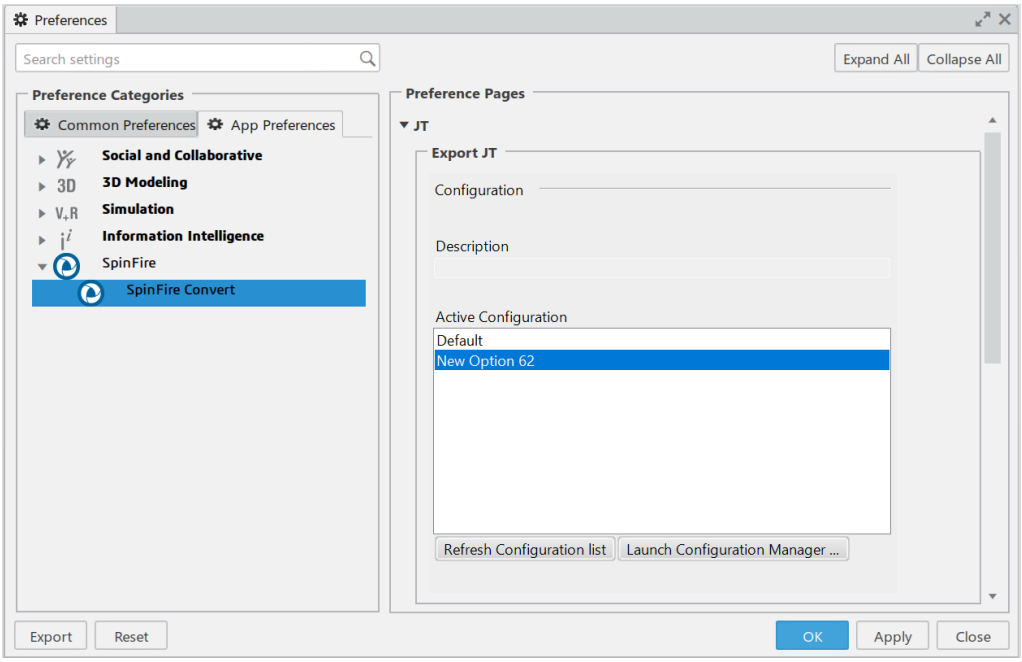


When all the relevant options have been applied, click on Exit to close the configuration manager window.

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



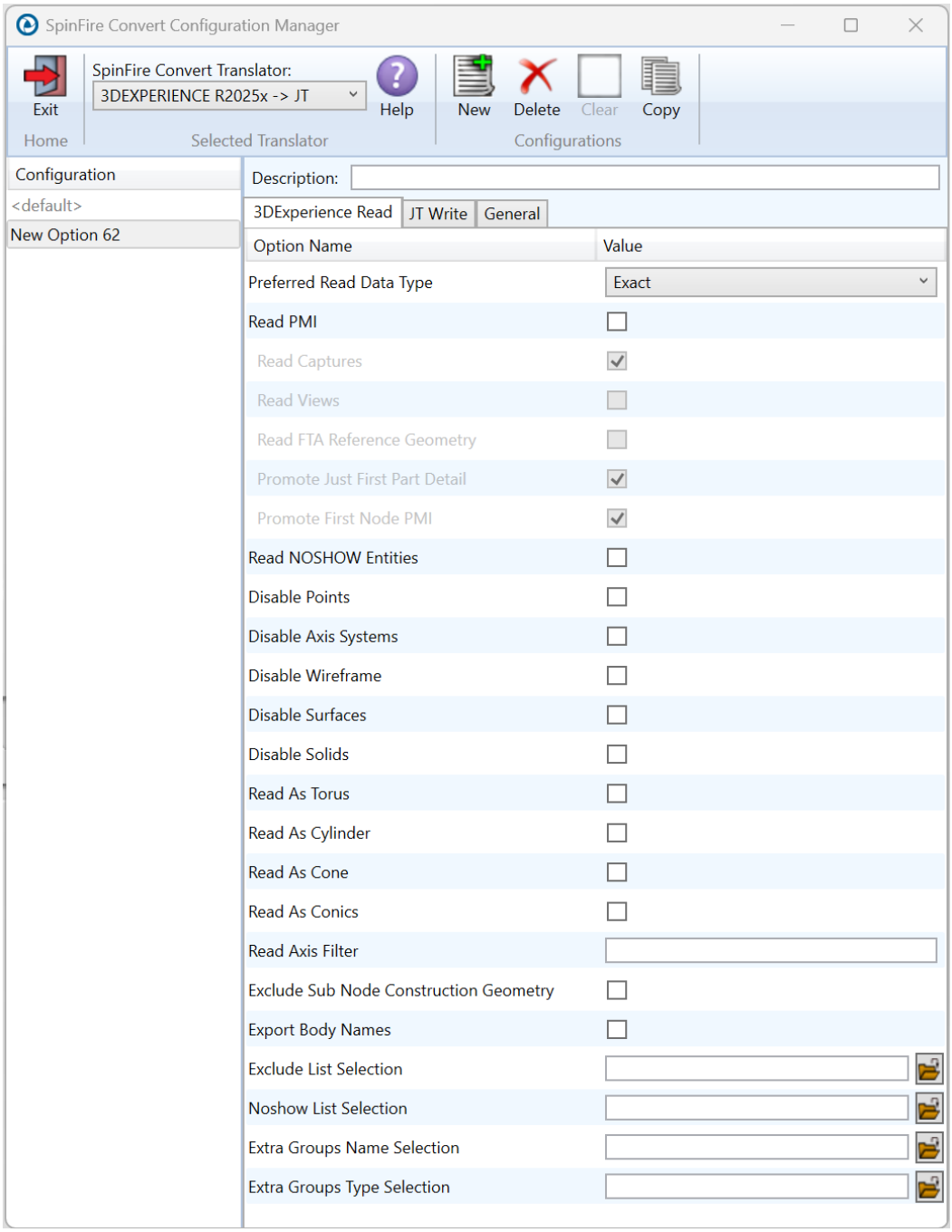
Select the required configuration to make it the active configuration. This will be highlighted in blue when selected.



Configuration Manager

3DExperience to JT

3DExperience Read



Each of the options displayed in the image above are described below:

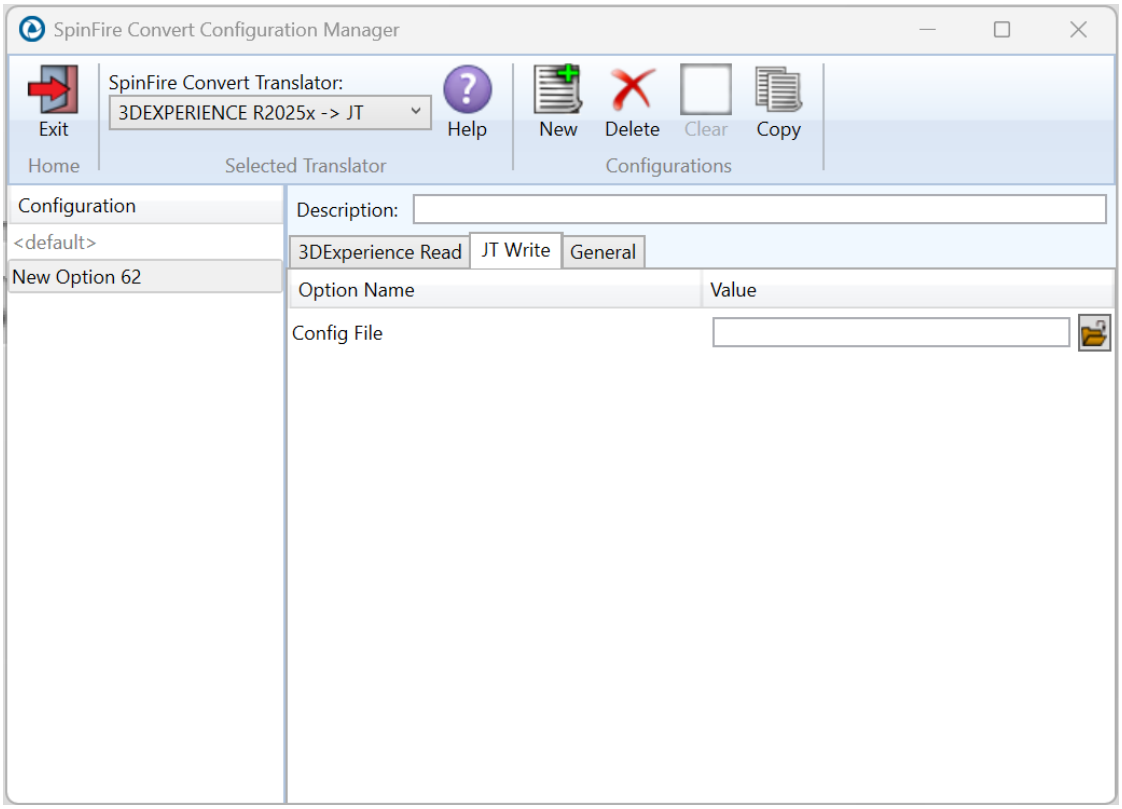
Option	Description
Preferred Read Data Type	The setting options are EXACT (default) or TESSELATED. Command Line Syntax: • *read_tess*
Read PMI	Enables PMI data read (Default is OFF.) Command Line Syntax: • *read_pmi_off – default* • *read_pmi – to turn on* Note – When ***'read_pmi'*** is enabled it also 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 ***'dont_fill_pmi_text'***
Read Captures	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. Command Line Syntax: • *read_captures - default* • *dont_read_captures – to turn off*
Read Views	A secondary argument to ***'Read PMI'*** and allows control over whether views are read as part of the process. Default is OFF. Views can ONLY be read when ***'Read PMI'*** is ON. Command Line Syntax: • *read_views – to turn on*
Read FTA Reference Geometry	A secondary argument to ***'Read PMI'*** and enables the reading of FTA Reference Geometry. Default is OFF. FTA Reference Geometry can ONLY be read when ***'Read PMI'*** is ON. Command Line Syntax: • *read_geometry – to turn on*
Promote Just First Part Detail	A secondary argument to ***'Read PMI'*** that enables the first node in the assembly in the specification tree to be promoted to the top node in the structure and subsequently read irrespective of what is contained in the node, i.e. geometry, PMI, Axis Systems etc. Default is ON when ***'Read PMI'*** is marked as ON. Can ONLY be read when ***'Read PMI'*** is ON. Command Line Syntax: • *promote_just_first_part_detail – default* • *promote_just_first_part_detail_off – to turn off*

Promote First Node PMI	A secondary argument to ***'Read PMI'*** that enables PMI on the first node in the assembly in the specification tree to be promoted to the top node in the structure and subsequently read. Default is ON when ***'Read PMI'*** is marked as ON. Can ONLY be read when ***'Read PMI'*** is ON. Command Line Syntax: • *promote_first_node_pmi – default* • *promote_first_node_pmi_off – to turn off* Note – Promote Just First Part detail will also need to be toggled on in order for this option to be applied.
Read NOSHOW Entities	Read any entities that are in NOSHOW. Default is not to read NOSHOW entities. Command Line Syntax: • *noshow*
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 Conics	Read surfaces generated from a Conic in analytical form (default is NURBS.) Command Line Syntax: • *read_conics*
Read Axis Filter	Enables a specified list of axis systems to be processed. Command Line Syntax: • *read_axis <value>* Where *<value>* is the name of the axis system. This can be formatted as per the examples below: • Axis System.1; – Only **'Axis System.1'** will be processed • *System.2; – Any axis system that includes **'System.2'** at the end of the name will be processed, i.e. Axis System.2 • Axis System*; – Any axis system that includes **'Axis System'** at the start of the name will be processed, i.e. Axis System.1, Axis System.2, Axis System.3, etc • *System*; – Any axis system that includes **'System'** anywhere in the name will be processed. Multiple axis systems can also be processed. This is formatted as per the example below: • Axis System.1;Axis System.2;Axis System.3;
Exclude Sub Node Construction Geometry	Excludes part level construction geometry from the output generated. Applying this option improves performance when exporting large assemblies. Command Line Syntax: • *subnode_cgeom_off*
Export Body Names	Maintains body names for parts that consist of multiple bodies. Command Line Syntax: • *body_names*
Exclude List Selection	Enables nodes to be excluded from the output generated when the corresponding node names have been added to a supplied list. This list will be in a .txt file format. Command Line Syntax: • *exclude_list <list file>* Where *<list file>* is the list of node names added to the text file. For example: Project Data External References

Noshow List Selection	Enables nodes that are in a noshow state to be included in the output generated when the corresponding node names have been added to a supplied list. This list will be in a .txt file format. Command Line Syntax: • *nowshow_list <list file>* Where *<list file>* is the list of node names added to the text file. For example: Geometrical Set.1 Geometrical Set.2
Extra Groups Name Selection	Adds an extra node per body below a node where the corresponding names have been specified in a supplied list. This list will be in a .txt file format. These extra nodes allow for finer control in what can be made visible / hidden in the output and results in better matching of Capture Views. Command Line Syntax: • *extra_groups_name_list <list file>* Where *<list file>* is the list of names added to the text file. For example: StiffenerSet_1 StiffenerSet_2
Extra Groups Type Selection	Adds an extra node per body below a node where its corresponding object types have been specified in a supplied list. This list will be in a .txt file format. These extra nodes allow for finer control in what can be made visible / hidden in the output and results in better matching of Capture Views. Command Line Syntax: • *extra_groups_type_list <list file>* Where *<list file>* is the list of object types added to the text file. For example: SldStiffenerSet SldPlateSet

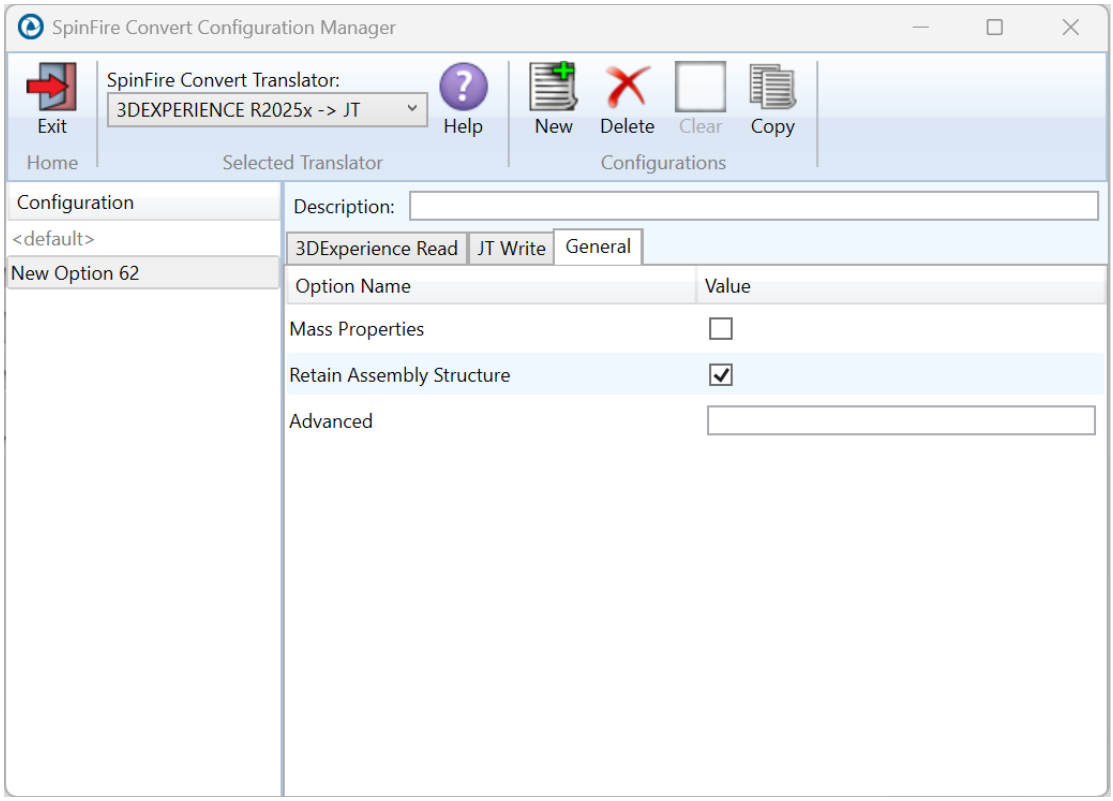
JT Write



Each of the options displayed in the Image above are described below:

Option	Description
Config File	Allows a JT configuration file to be specified. "<TS_INST>/etc/tess.config" is the default. (See Appendix A for config file options.) Command Line Syntax: • *-z <path to JT configuration file>*

General

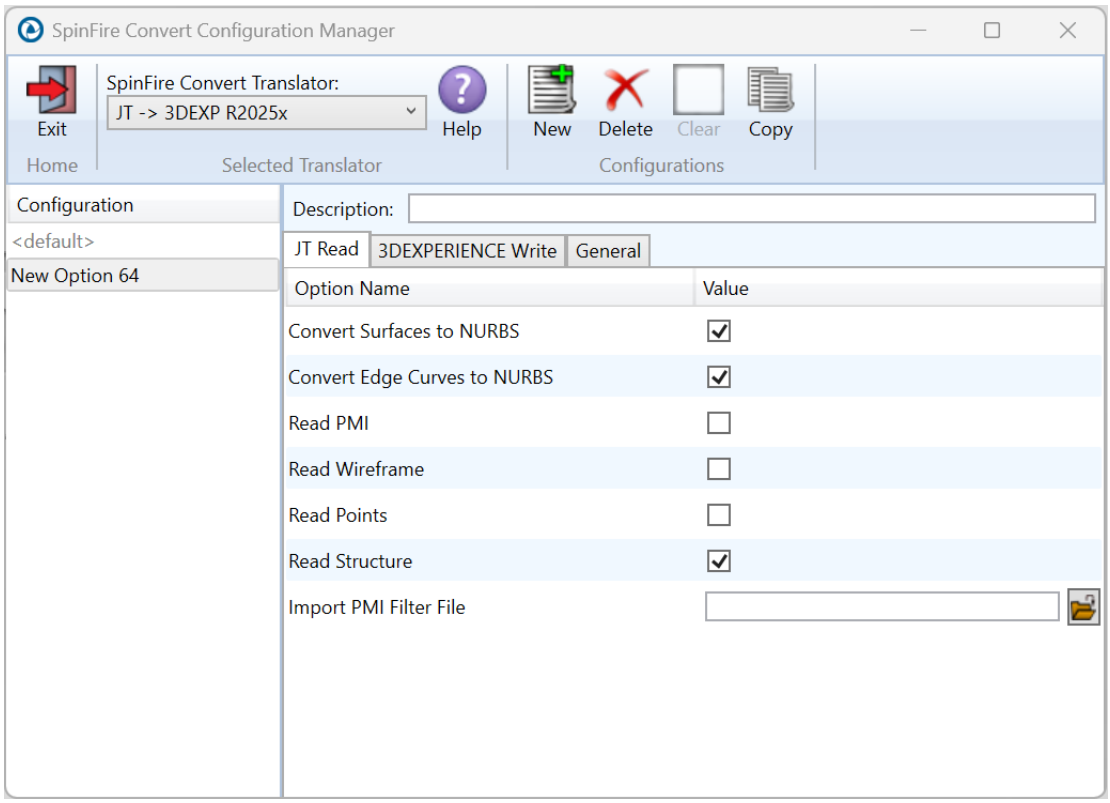


Each of the options displayed in the image above are described below:

Option	Description
Mass Properties	Allows mass property information to be read. Command Line Syntax: • *mprops*
Retain Assembly Structure	Enables Assembly Structure to be retained. (Default is ON.) Disabling this option will remove all assembly structure and collapse **ALL** geometry into a single selectable object. Command Line Syntax: • *off_ditto – to turn off*
Advanced	Allows any advanced arguments to be added to the configuration manager and applied during the translation.

JT to 3DExperience

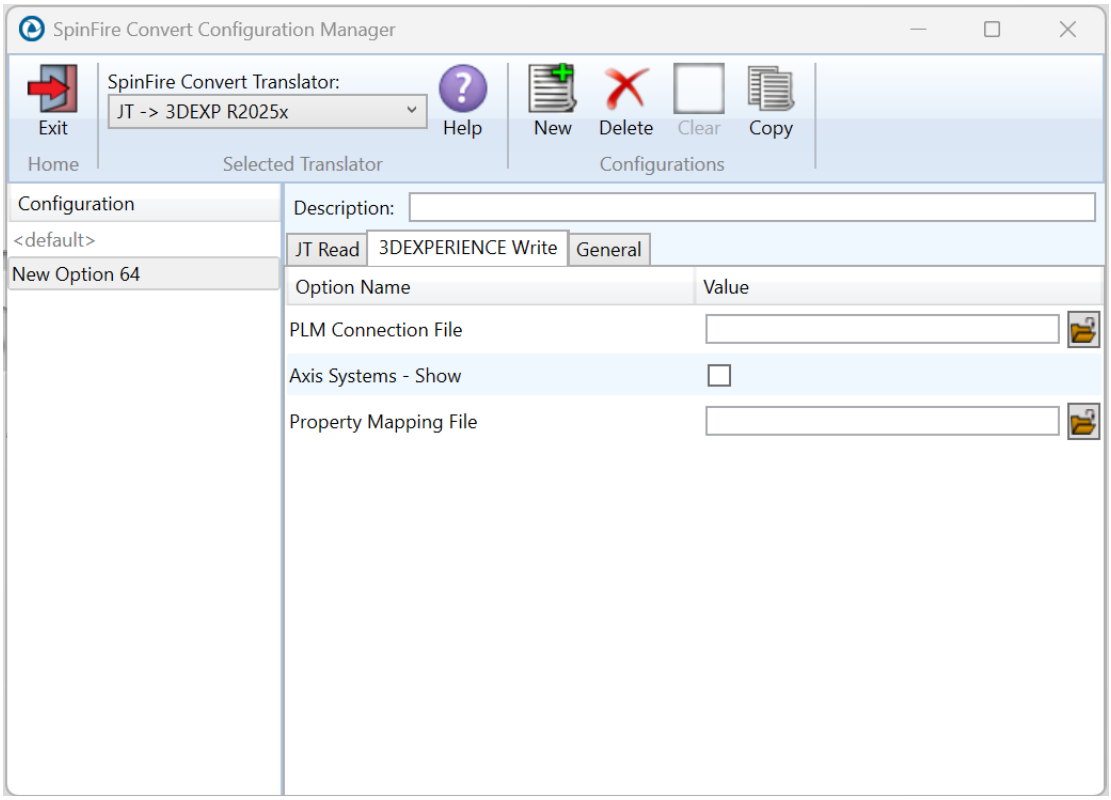
JT Read



Each of the options displayed in the image above are described below:

Option	Description
Convert Surfaces to NURBS	Read XT Brep surfaces as NURBS surfaces (else read in native form). Default is ON.
Convert Edge Curves to NURBS	Read XT Brep edge curves as NURBS curves (else read in native Form). Default is ON.
Read PMI	Reads 3D PMI. Default is OFF.
Read Wireframe	Reads JT wireframe data. Default is OFF.
Read Points	Reads JT point data. Default is OFF.
Read Structure	Read assembly tree structure. Default is ON.
Import PMI filter file	Supply a PMI filter file. "<TS_INST>/data/jt/jt_pmi_filter.txt" is the default. ***(See Appendix D for more information on PMI Filter Files.)***

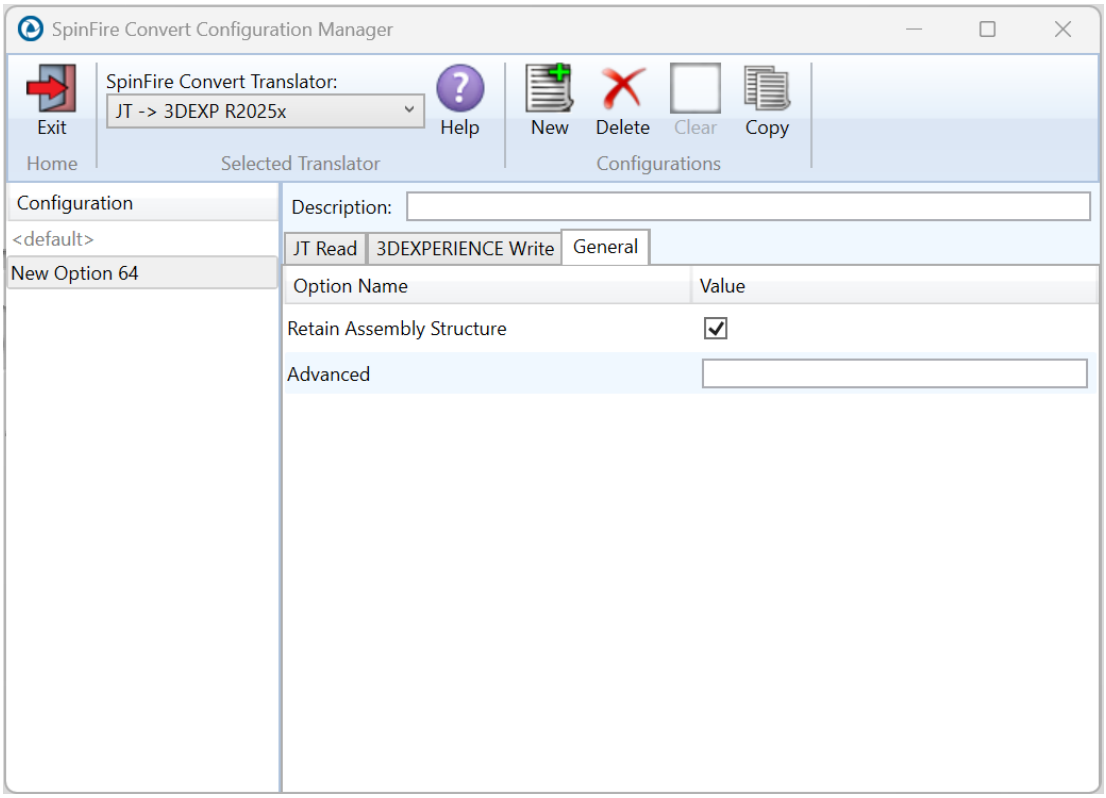
3DExperience Write



Each of the options displayed in the image above are described below:

Option	Description
PLM Connection File	The PLM Connection file is one of the xml files defining the connection parameters to the PLM database. It is required to create a connection and logging into a PLM session programmatically.
Axis Systems - Show	Show all the axis systems on the imported object in 3DExperience.
Property Mapping File	Map CAD properties using a mapping file. (***See Appendix C for more information on Property Mapping Files.)***

General



Each of the options displayed in the image above are described below:

Option	Description
Retain Assembly Structure	Enables Assembly Structure to be retained. (Default is ON.) Disabling this option will remove all assembly structure and collapse **ALL** geometry into a single selectable object.
Advanced	Allows any advanced arguments to be added to the configuration manager and applied during the translation.

Translating in Batch using CATUTIL – DataExchangePLMBatch

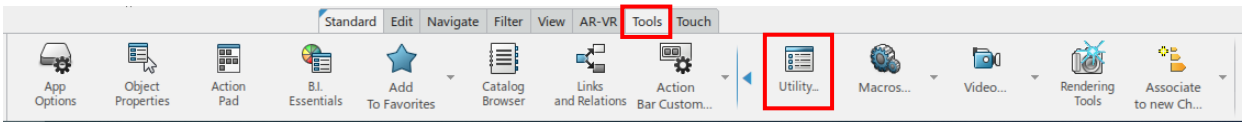
Launching DataExchangePLMBatch

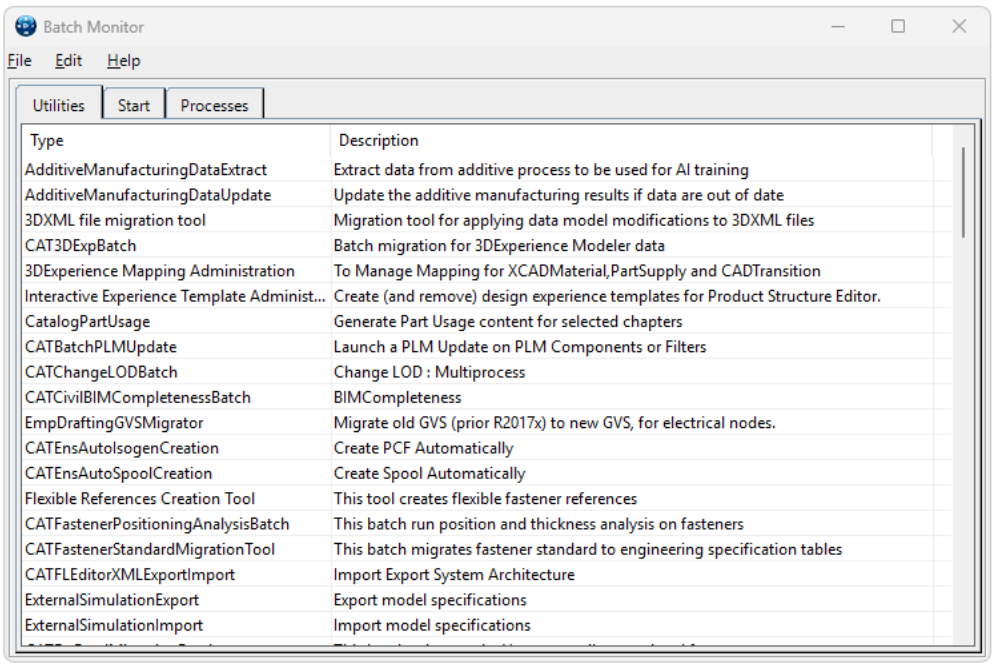
The Dassault Systemes CATUTIL Batch Monitor tool can be launched with the correct SpinFire Convert environment using the scripts provided in the Translator installation at:

<installation_directory>\bin\start_CATUTIL_<version>_JT.cmd

Where *<version>* is the version of 3DExperience that you have installed – e.g. 2023x, 2024x, 2025x.

It can also be started from the Tools>Utility workbench within an Interactive 3DExperience session.





In some cases, it may be desirable to ignore the interactive settings while translating using DataExchangePLMBatch.

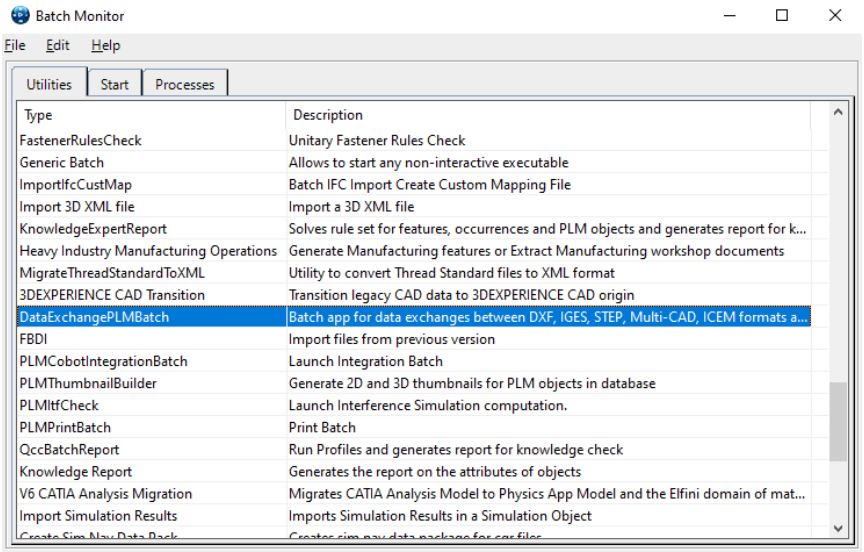
This can be achieved by setting the following variable in the SpinFire Convert CATEnv file:

TS_IGNORE_JT_CATSETTINGS=1

Batch Import from JT

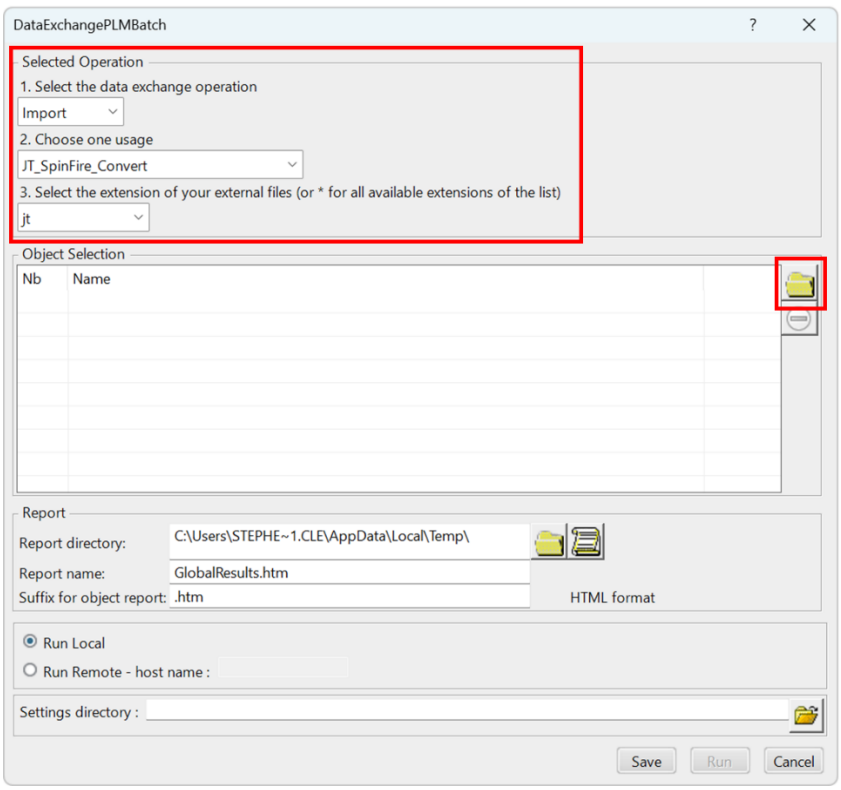
Here is an example showing what is required in preparation for a JT file selection and batch import.

Double click on **DataExchangePLMBatch** in the Batch Monitor window.

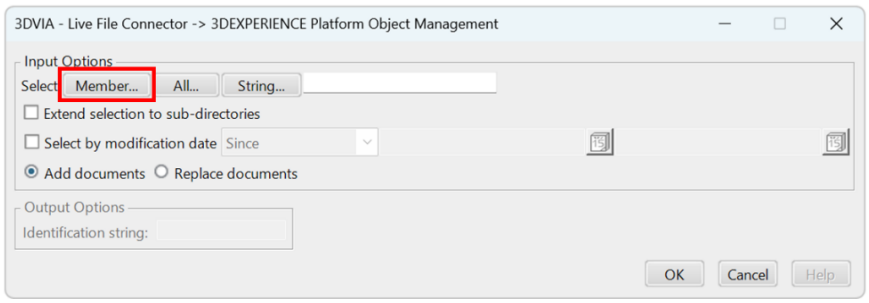


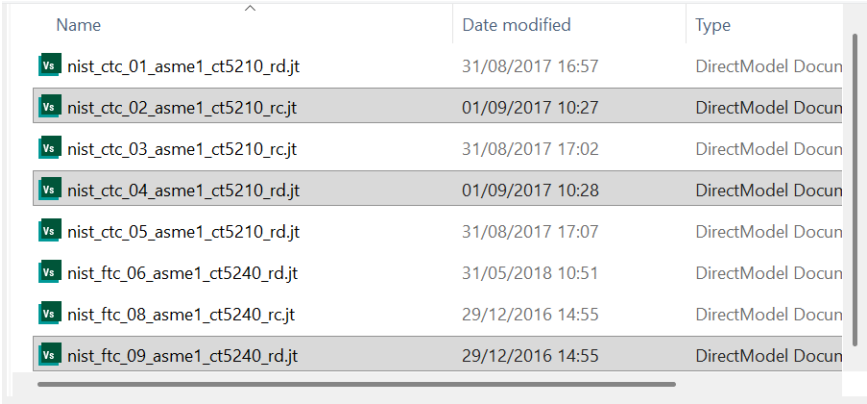
Ensure that ***'Import'*** is selected as the data exchange operation, ***'JT_SpinFire_Convert'*** is selected as the usage and ***'jt', 'plmxml' or 'stpx'*** is selected as the extension.

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



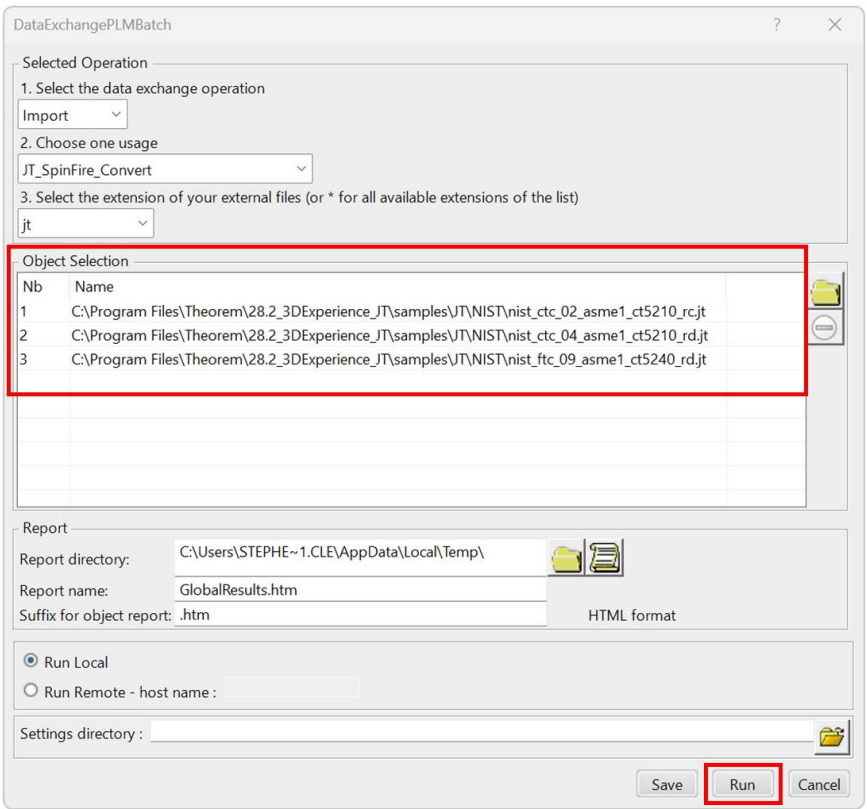
In the subsequent window displayed, add the products by clicking on Member and selecting the required JT files.



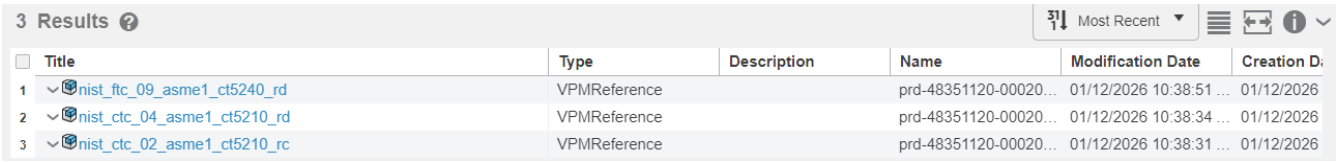


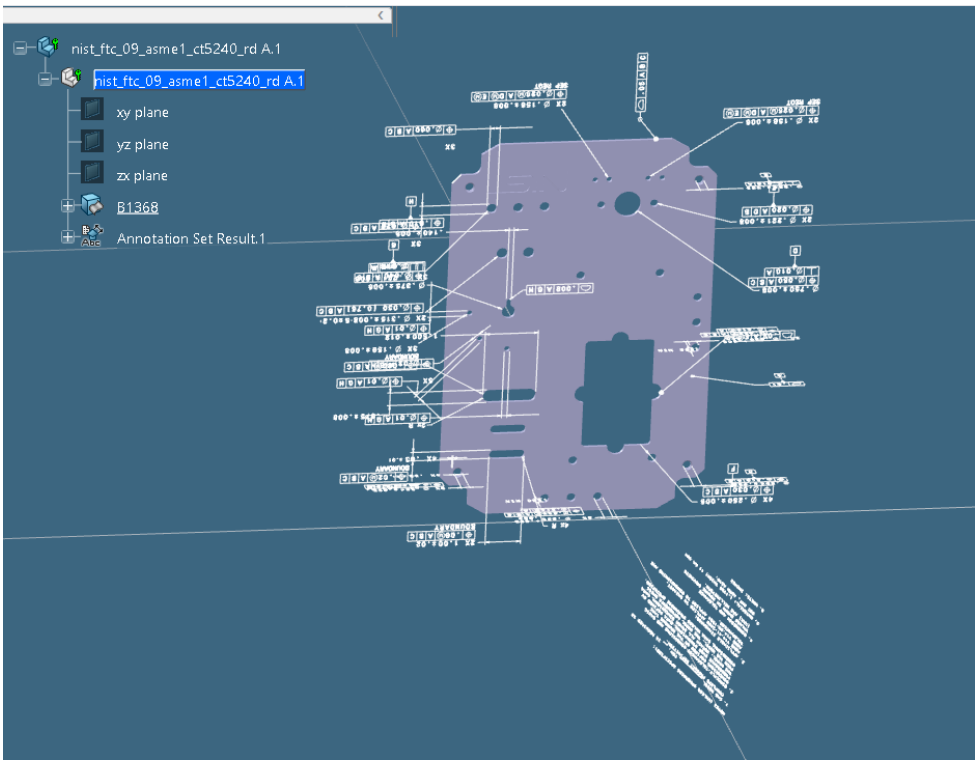
Click OK to add them to the Object Selection field.

Click Run to initiate the Batch Import.



Imported data saved into the 3DExperience database.

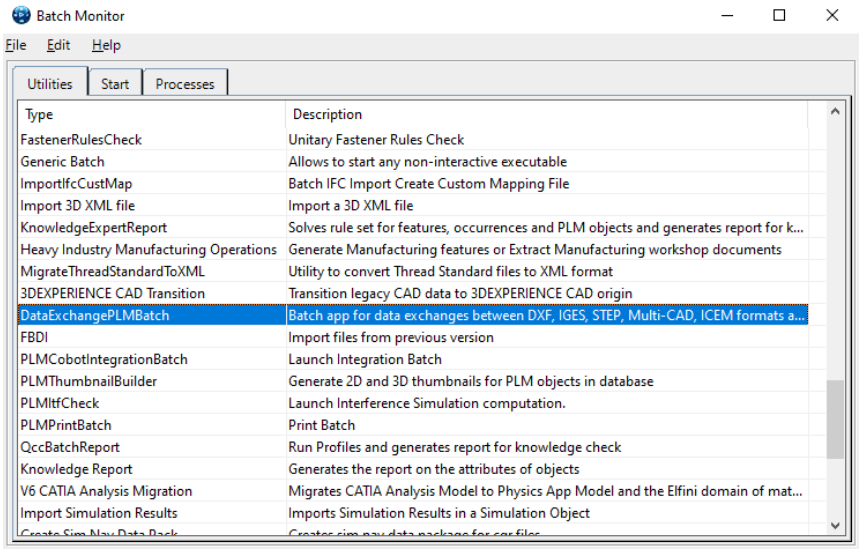




Batch Export to JT

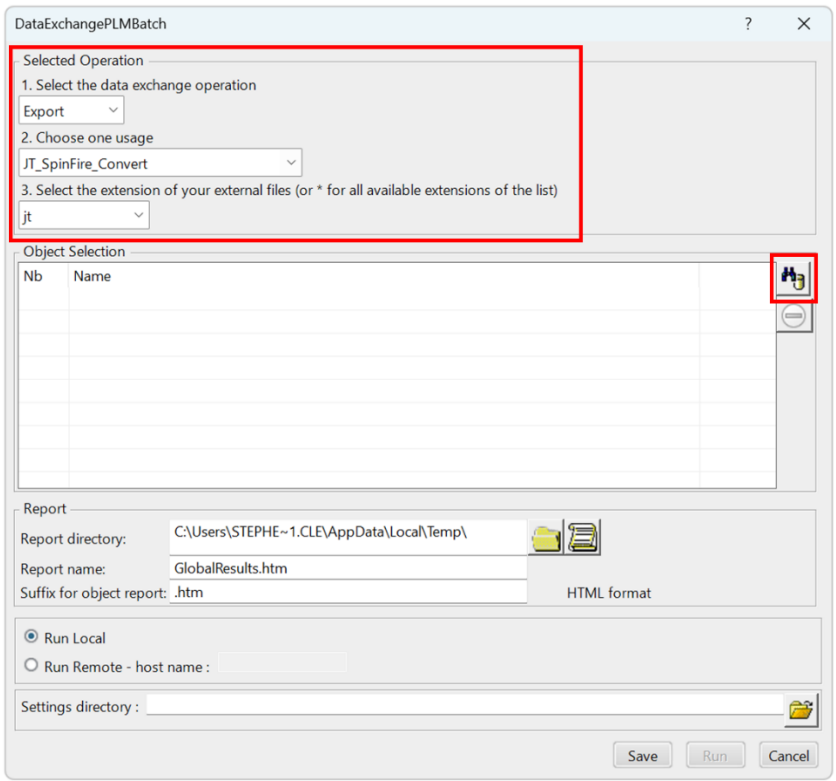
Here is an example showing what is required in preparation for a 3DExperience database selection and batch export.

Double click on **DataExchangePLMBatch** in the Batch Monitor window.

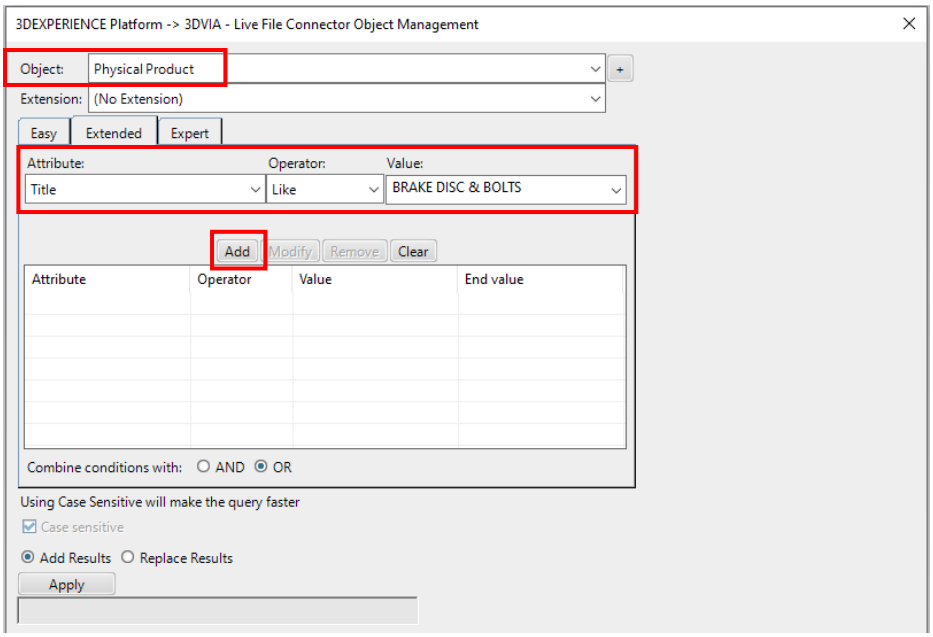


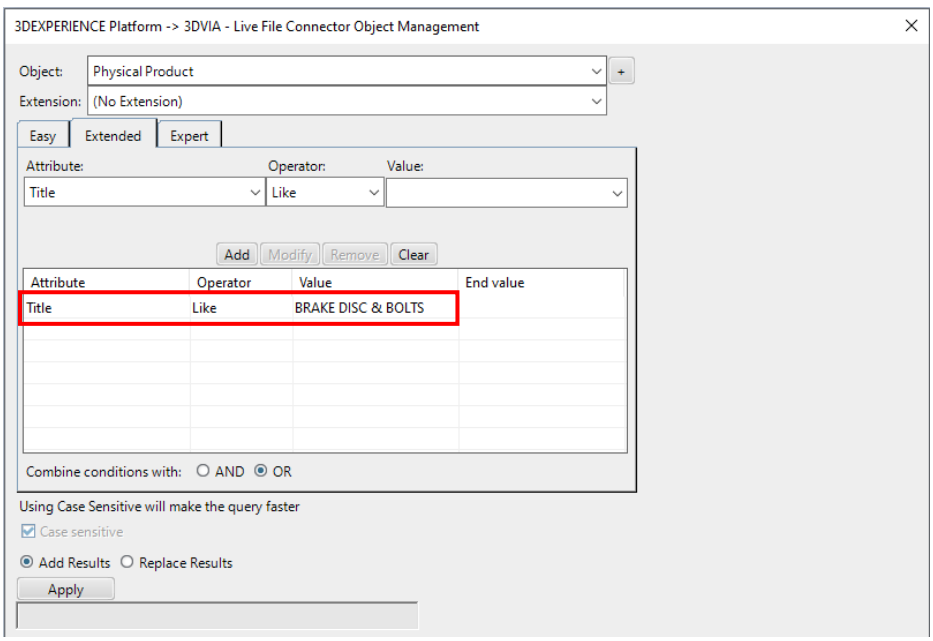
Ensure that ***'Export'*** is selected as the data exchange operation, ***'JT_SpinFire_Convert'*** is selected as the usage and ***'jt', 'plmxml' or 'stpx'*** is selected as the extension.

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



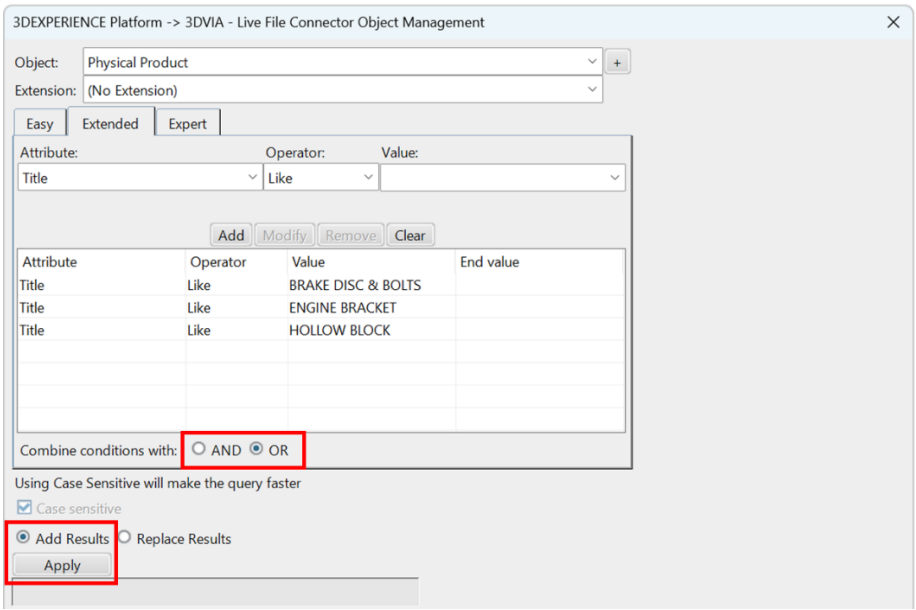
In the subsequent window displayed, add the required products using the correct attribute, operator and value ensuring that the correct object type has been selected. (E.g. Physical Product). Where required use an Asterix in the value field to identify multiple parts or assemblies.

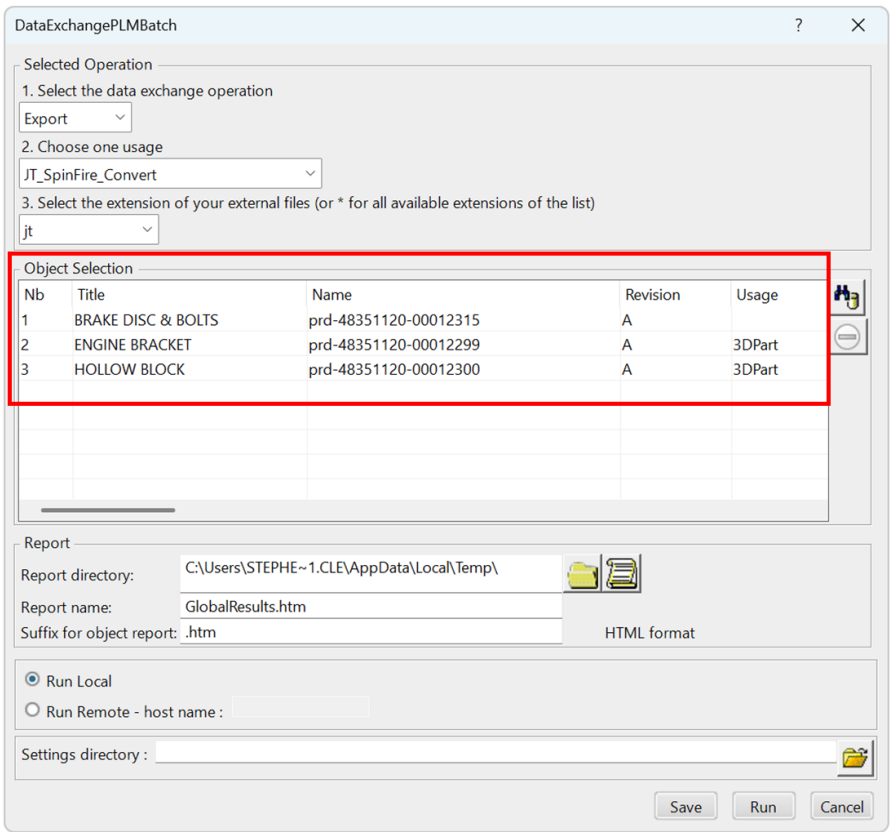




For multiple products ensure that the **OR** condition has been selected.

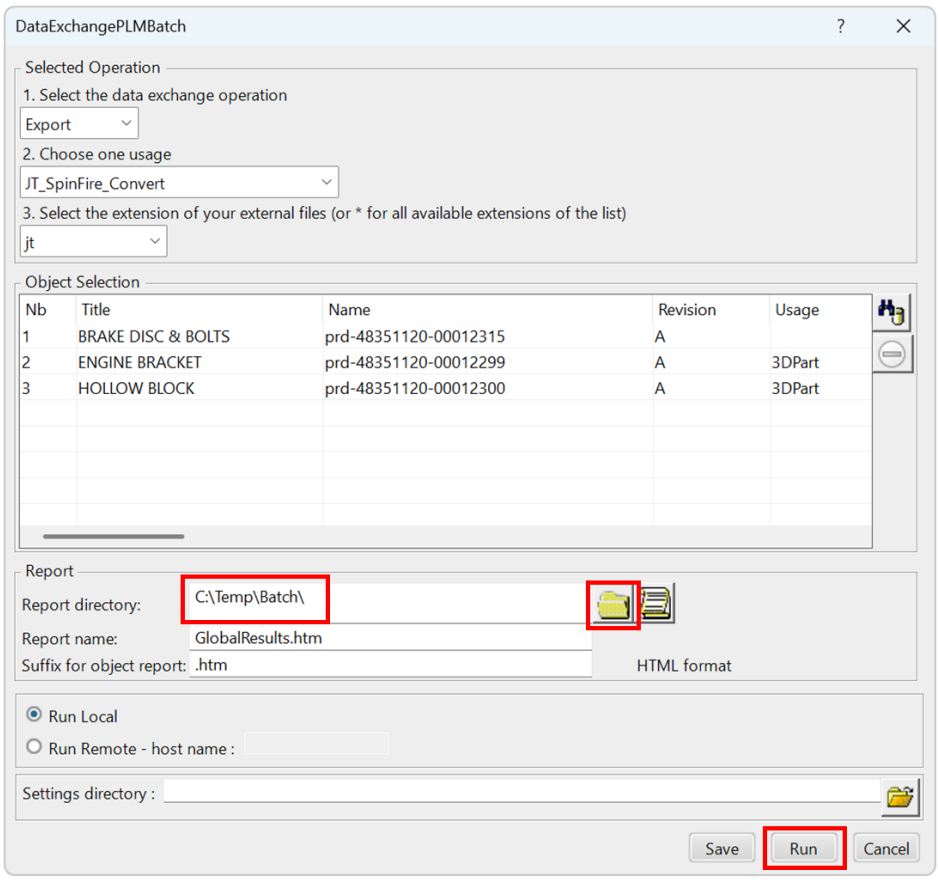
Select Add results and click Apply to add the list of products to the Object Selection field.

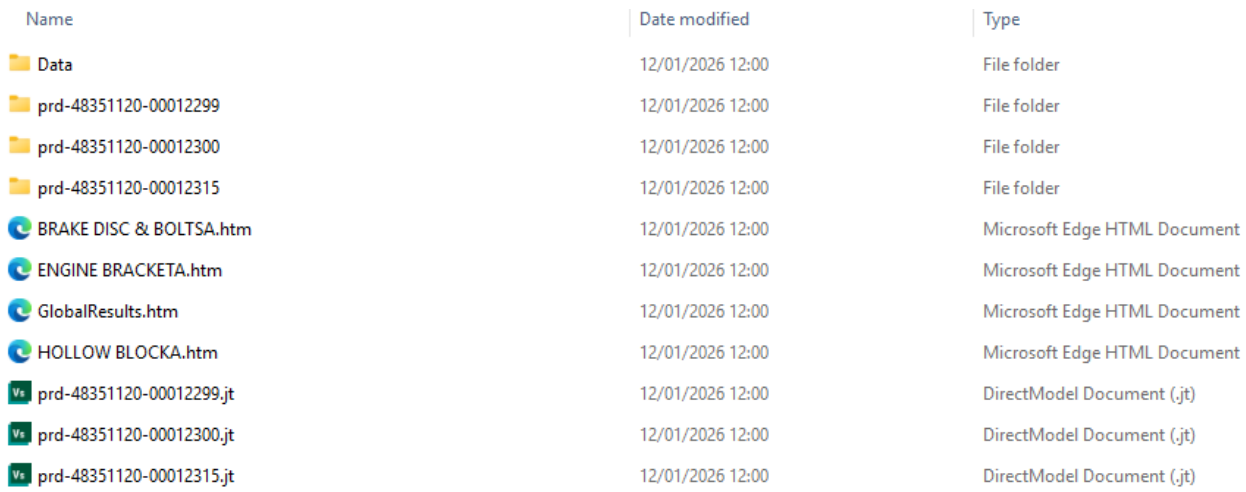




To change the location where both the report and the JT files will be saved, select the  icon next to the Report directory field, then select the required directory.

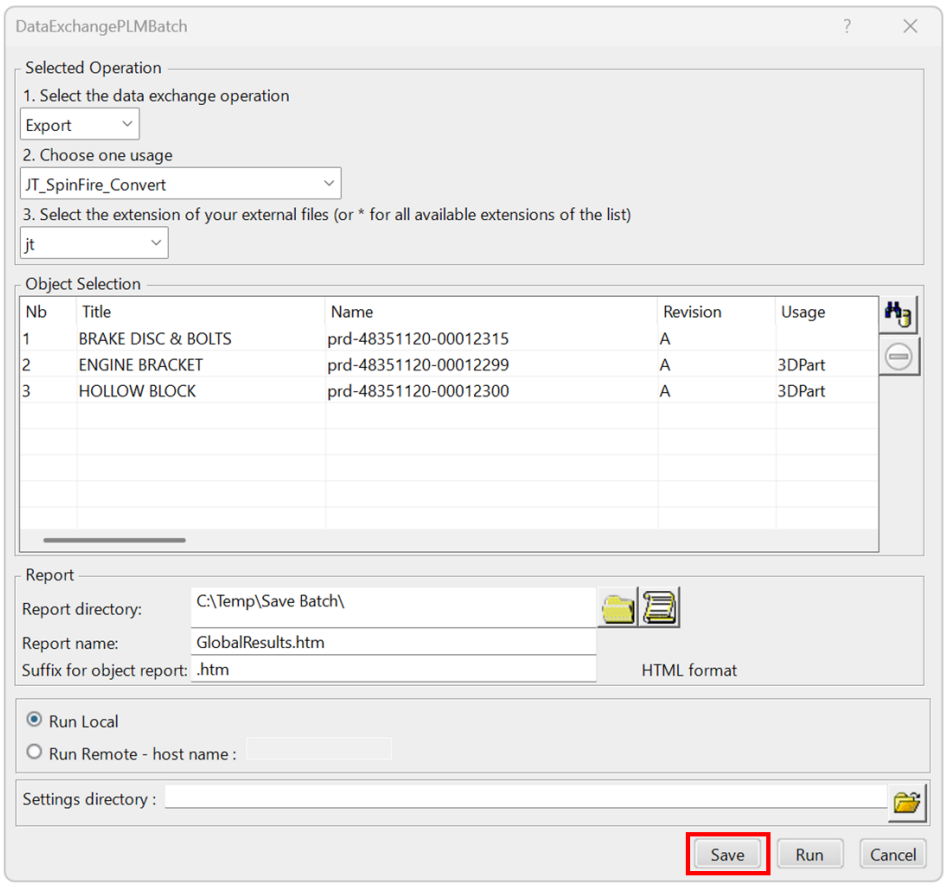
Click Run to initiate the Batch Export.



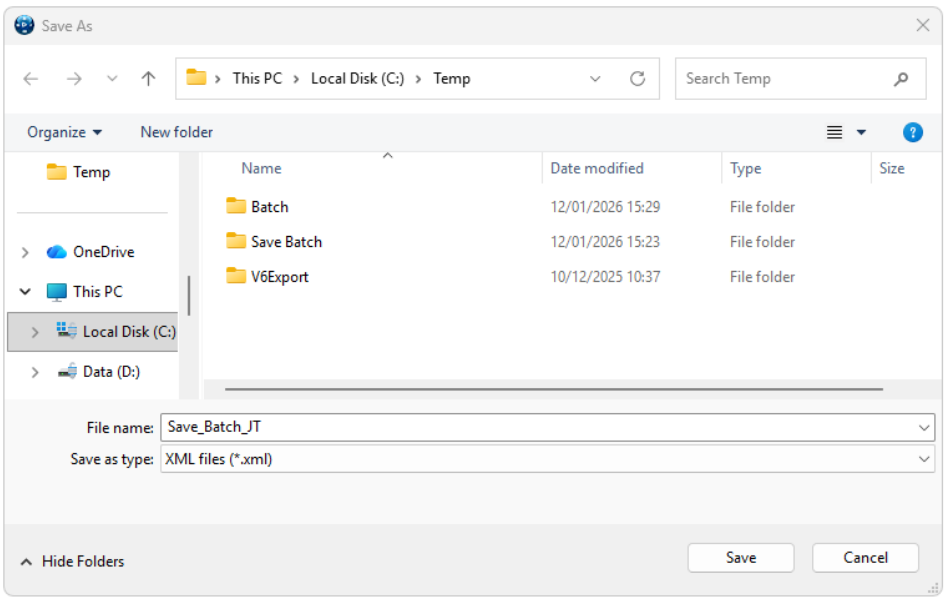


Running a Saved Batch File

It is also possible to run a saved batch file on the command line. To do this click **Save** instead of Run in the DataExchangePLMBatch window.



In the Save As pop-up window displayed, enter a file name, select the required directory where the batch file will be saved, then click Save.



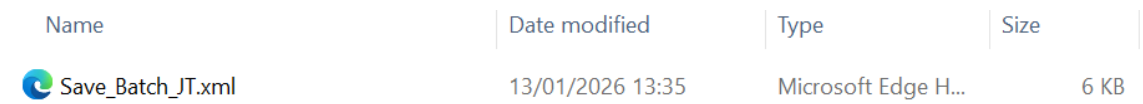
In order to run a saved batch file on the command line the following is required:

<installation_directory>\bin\start_CATBatchStarter_<version>_JT.cmd <input file>

Where *<installation_directory>* is the SpinFire Convert software installation directory.
Where *<version>* is the version of 3DExperience that you have installed – e.g. 2023x, 2024x, 2025x.

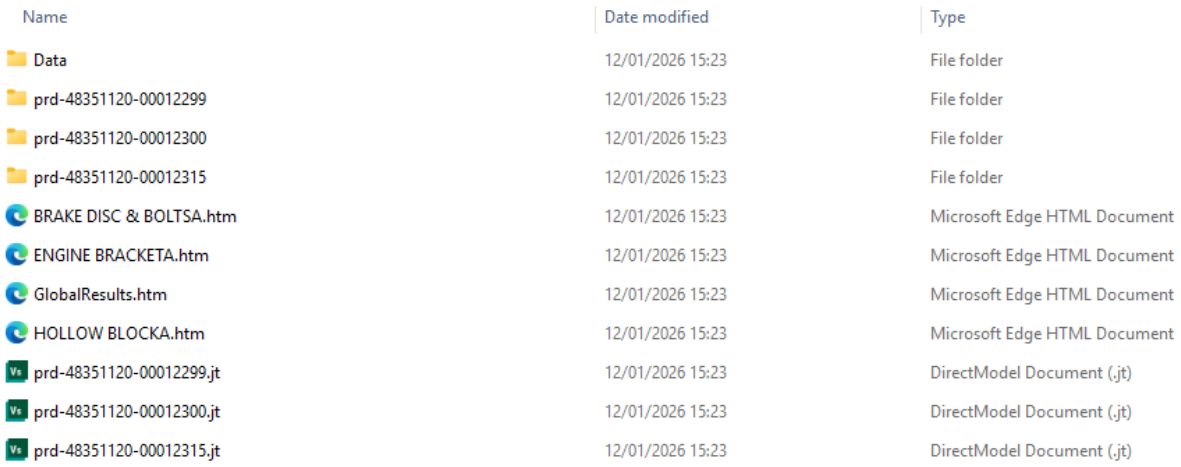
<input_file>

Is the saved batch file. This will be in an XML file format.



Name	Date modified	Type	Size
Save_Batch_JT.xml	13/01/2026 13:35	Microsoft Edge H...	6 KB

```
C:\Users\stephen.clews>"C:\Program Files\Theorem\28.2_3DExperience_JT\bin\start_CATBatchStarter_2025x_JT.cmd"
C:\Temp\Save_Batch_JT.xml
```



Name	Date modified	Type
Data	12/01/2026 15:23	File folder
prd-48351120-00012299	12/01/2026 15:23	File folder
prd-48351120-00012300	12/01/2026 15:23	File folder
prd-48351120-00012315	12/01/2026 15:23	File folder
BRAKE DISC & BOLTSA.htm	12/01/2026 15:23	Microsoft Edge HTML Document
ENGINE BRACKETA.htm	12/01/2026 15:23	Microsoft Edge HTML Document
GlobalResults.htm	12/01/2026 15:23	Microsoft Edge HTML Document
HOLLOW BLOCKA.htm	12/01/2026 15:23	Microsoft Edge HTML Document
prd-48351120-00012299.jt	12/01/2026 15:23	DirectModel Document (.jt)
prd-48351120-00012300.jt	12/01/2026 15:23	DirectModel Document (.jt)
prd-48351120-00012315.jt	12/01/2026 15:23	DirectModel Document (.jt)

Please note the JT files will be saved in the location specified in the Report Directory in the DataExchangePLMBatch window.

Changing the Options Selected for a Batch Export / Import

The default options selected for a batch export and import can be modified within the **xcad_jt_opts.txt** file and **jt_xcad_opts.txt** file respectively. Both files are available in the following location.

<installation_directory>\data\jt

Within the **xcad_jt_opts.txt** file a number of arguments are displayed, these include common arguments such as read_pmi, disable_points, disable_wireframe and disable_axes which are all turned off by default. If an argument has the character (!) at the start of the line then this means that the argument is turned off.

```
14  !!!!!!!!!!!!!!!!!!!
15  ! V6 Read options
16  !!!!!!!!!!!!!!!!!!!
17  !disable_points
18  !disable_wireframe
19  !disable_surfaces
20  !disable_solids
21  !disable_axes
22  !read_pmi
23  ! To read PMI rendered as polylines only, uncomment the following line.
24  !dont_fill_pmi_text
25  !noshow
26  !body_names
```

To turn on a particular argument remove the (!) character from that particular line. Consequently, to turn off an argument add the (!) character to the start of the line. Please note that the user will need to be in administrator mode in order to make the change.

```
14  !!!!!!!!!!!!!!!!!!!
15  ! V6 Read options
16  !!!!!!!!!!!!!!!!!!!
17  disable_points
18  disable_wireframe
19  !disable_surfaces
20  !disable_solids
21  disable_axes
22  read_pmi
23  ! To read PMI rendered as polylines only, uncomment the following line.
24  !dont_fill_pmi_text
25  !noshow
26  !body_names
```

Within the **jt_xcad_opts.txt** file a number of arguments are also displayed, these include read_points, read_wire_frame and show_axis_system which are all turned off by default. If an argument has the character (*) at the start of the line then this means that the argument is turned off.

```
49  *------------------- Entity type filtering ---------------------------
50  * By default, points and wireframe curve entities are not read.
51  * If point entities are required, uncomment the following line
52  *read_points
53  *
54  * If wireframe entities are required, uncomment the following line
55  *read_wire_frame
56  *
57  * Ignore the jt subnode property and read underlying (leaf) structure
58  *no_subnode
59  *
60  * Disable read of 3D VIEWs
61  *NO_READ_VIEWS
62  *
63  * Axis systems will by default be created in hidden visibility space,
64  * to create Axis systems in shown visibility space, uncomment the following line
65  *show_axis_system
66  *
```

Once again, to turn on a particular argument remove the (*) character from that particular line. Consequently, to turn off an argument add the (*) character to the start of the line. Please note that the user will need to be in administrator mode in order to make the change.

```
49  *------------------- Entity type filtering ---------------------------
50  * By default, points and wireframe curve entities are not read.
51  * If point entities are required, uncomment the following line
52  read_points
53  *
54  * If wireframe entities are required, uncomment the following line
55  read_wire_frame
56  *
57  * Ignore the jt subnode property and read underlying (leaf) structure
58  *no_subnode
59  *
60  * Disable read of 3D VIEWs
61  *NO_READ_VIEWS
62  *
63  * Axis systems will by default be created in hidden visibility space,
64  * to create Axis systems in shown visibility space, uncomment the following line
65  show_axis_system
66  *
```

Changing the JT Configuration Selected for a Batch Export

The JT configuration file selected for a batch export can also be modified within the **xcad_jt_opts.txt** file.

To change the JT configuration, simply remove the existing file path displayed and replace it with the new file path. The JT configuration argument will be displayed under the JT Write options section in the xcad_jt_opts.txt file. Note – ensure that the -z character is not removed.

```
58  !!!!!!!!!!!!!!!!!!!
59  ! JT Write options
60  !!!!!!!!!!!!!!!!!!!
61  -z "%TS_INST%\etc\tess.config"
62  !
```

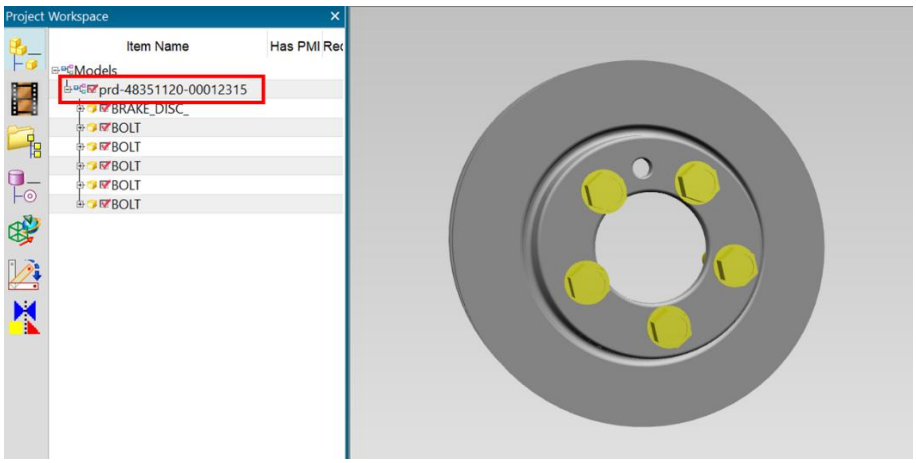
```
58  !!!!!!!!!!!!!!!!!!!
59  ! JT Write options
60  !!!!!!!!!!!!!!!!!!!
61  -z "C:\Program Files\Theorem\26.2_3DExperience_JT\etc\mono.config"
62  !
```

These changes will consequently mean that the user specified JT configuration will be used instead of the default JT configuration when translating via DataExchangePLMBatch. Please note that the user will need to be in administrator mode in order to make the change.

Changing the Name of the Top Node in the JT Output

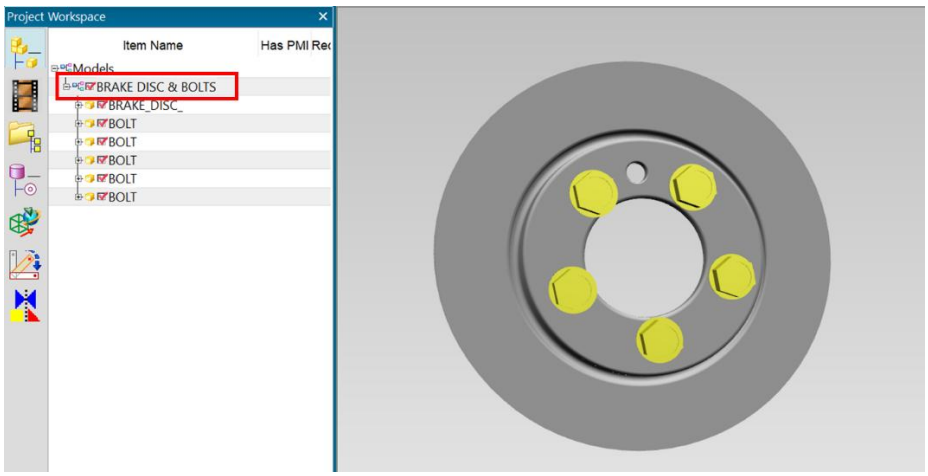
When exporting data using DataExchangePLMBatch, ensure that the design_name option has been turned on in the **xcad_jt_opts.txt** file. If this option is turned off then the top node displayed in the JT output will be named the object name displayed in 3DExperience, (this is a 3DExperience generated name/number). The design_name option is turned **OFF** by default.

```
82  !
83  ! To use pmi glyphs comment the following line
84  pmi_glyphs_off
85  ! To use pmi polygons uncomment the following line
86  pmi_polygons
87  ! To name route node from DESIGN ent. Enabling this option forces a Monolythic JT
88  !design_name
89  !
```



If the design_name option is turned on then the top node displayed in the JT output will be named the object title displayed in 3DExperience, (this is a user generated name/number). Turning on the design_name option will also generate a monolithic JT output. Removing the (!) character will turn this option on. Please note that the user will need to be in administrator mode in order to make the change.

```
82  !
83  ! To use pmi glyphs comment the following line
84  pmi_glyphs_off
85  ! To use pmi polygons uncomment the following line
86  pmi_polygons
87  ! To name route node from DESIGN ent. Enabling this option forces a Monolythic JT
88  design_name
89  !
```



Changing the JT File Name

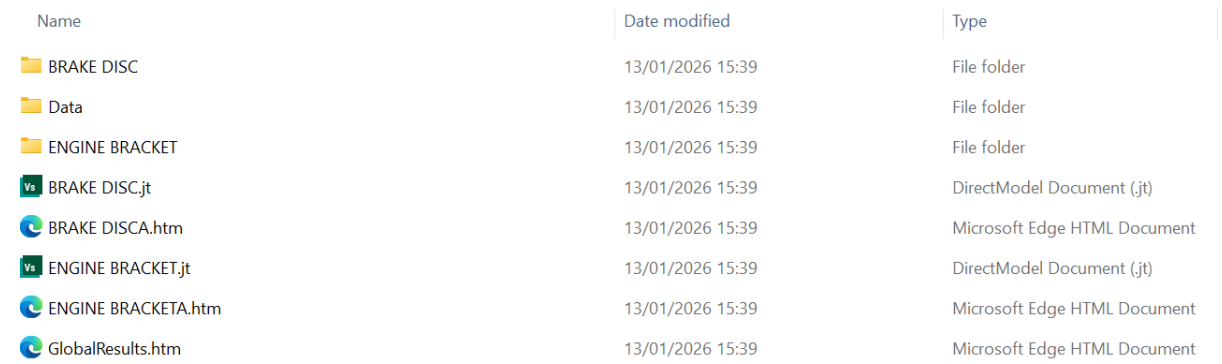
When exporting data using DataExchangePLMBatch, ensure that the part_naming #PN# option has been turned on in the **xcad_jt_opts.txt** file. If this option is turned off then the JT file will be named the object name displayed in 3DExperience, (this is a 3DExperience generated name/number). The part_naming #PN# option is turned **OFF** by default.

```
! Alter the output part name part_naming <String>
! Where <String> can contain mapping for new name where
! Product
!    #PN# - Part Number
!    #NO# - Nomenclature
!    #RE# - Revision
!    #SO# - Source
!    #DF# - Definition
!    #DR# - Description
! Link to Reference
!    #SRP# - Short Reference Path
! Component
!    #IN# - Instance Name
!    #DI# - Description
! Shape
!    #SN# - Shape Name
!    #EN# - Enviroment
!    #SD# - Short Description
!    #SC# - Source
! Or
!    #@GCO ATTRIBUTE NAME@#
!
! Example
! part_naming
! #PN#
```

Name	Date modified	Type
Data	13/01/2026 15:33	File folder
prd-48351120-00012298	13/01/2026 15:33	File folder
prd-48351120-00012299	13/01/2026 15:34	File folder
BRAKE DISCA.htm	13/01/2026 15:34	Microsoft Edge HTML Document
ENGINE BRACKETA.htm	13/01/2026 15:34	Microsoft Edge HTML Document
GlobalResults.htm	13/01/2026 15:34	Microsoft Edge HTML Document
prd-48351120-00012298.jt	13/01/2026 15:33	DirectModel Document (.jt)
prd-48351120-00012299.jt	13/01/2026 15:34	DirectModel Document (.jt)

If the part_naming #PN# option is turned on then the JT file will be named the object title displayed in 3DExperience, (this is a user generated name/number). Removing the (!) character from both the part_naming and #PN# lines will turn this option on. Please note that the user will need to be in administrator mode in order to make the change.

```
170  ! Alter the output part name part_naming <String>
171  ! Where <String> can contain mapping for new name where
172  ! Product
173  !    #PN# - Part Number
174  !    #NO# - Nomenclature
175  !    #RE# - Revision
176  !    #SO# - Source
177  !    #DF# - Definition
178  !    #DR# - Description
179  ! Link to Reference
180  !    #SRP# - Short Reference Path
181  ! Component
182  !    #IN# - Instance Name
183  !    #DI# - Description
184  ! Shape
185  !    #SN# - Shape Name
186  !    #EN# - Enviroment
187  !    #SD# - Short Description
188  !    #SC# - Source
189  ! Or
190  !    #@GCO ATTRIBUTE NAME@#
191  !
192  ! Example
193  part_naming
194  #PN#
```



Name	Date modified	Type
BRAKE DISC	13/01/2026 15:39	File folder
Data	13/01/2026 15:39	File folder
ENGINE BRACKET	13/01/2026 15:39	File folder
BRAKE DISC.jt	13/01/2026 15:39	DirectModel Document (.jt)
BRAKE DISCA.htm	13/01/2026 15:39	Microsoft Edge HTML Document
ENGINE BRACKET.jt	13/01/2026 15:39	DirectModel Document (.jt)
ENGINE BRACKETA.htm	13/01/2026 15:39	Microsoft Edge HTML Document
GlobalResults.htm	13/01/2026 15:39	Microsoft Edge HTML Document

Please note that different attributes can also be used to name the output file, for example part_naming #DR# will name the JT file the object description.

```
! Alter the output part name part_naming <String>
! Where <String> can contain mapping for new name where
! Product
!    #PN# - Part Number
!    #NO# - Nomenclature
!    #RE# - Revision
!    #SO# - Source
!    #DF# - Definition
!    #DR# - Description
! Link to Reference
!    #SRP# - Short Reference Path
! Component
!    #IN# - Instance Name
!    #DI# - Description
! Shape
!    #SN# - Shape Name
!    #EN# - Enviroment
!    #SD# - Short Description
!    #SC# - Source
! Or
!    #@GCO ATTRIBUTE NAME@#
!
! Example
part_naming
#DR#
```

Translating on the Command Line

It is possible to run a 3DExperience to JT translation on the command line, however as all CAD data is saved in the 3DExperience database an XML file will need to be used as the input file instead of a CAD file. This XML file contains all the necessary information to locate, open and translate the CAD data. This is explained in more detail below. In order to run a translation on the command line the following is required:

<installation_directory>\bin\catia6r<revision>_ jt.cmd <input_file> -o <output_file> -z <config_file> <options>

Where *<installation_directory>* is the SpinFire Convert software installation directory.
Where *<version>* is the version of 3DExperience that you have installed – e.g. 2023x, 2024x, 2025x.

<input_file>

Is an xml file defining access to a specific object in the 3DExperience database.

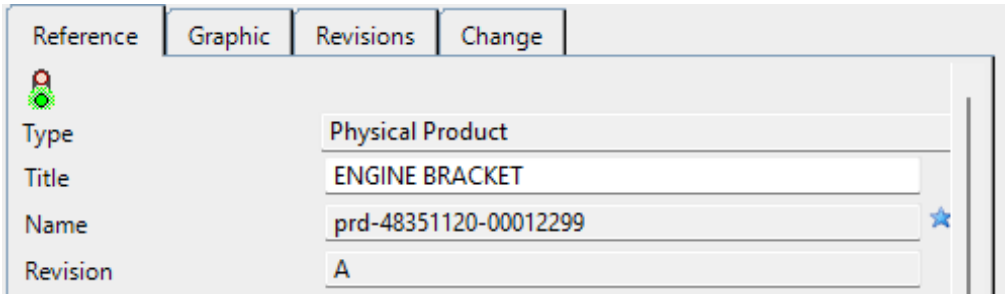
This file provides user login details (V6R2013x) or a Login Ticket (2015x onwards) ***(see appendix F),*** to a specified Enovia repository plus a set of attribute names and values which will uniquely identify the input object required for the conversion. These are highlighted in green in the example below and will need changing to the object and revision being translated. The lines highlighted in yellow should be modified with 'your' login details which are required to access Enovia. Example XML files can be found in the ***<installation_directory >\samples\3dexperience*** folder. These can be modified and saved on your local drive.

Here is an example of the xml input file for 3DExperience

```
<?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="https://3dspace-2025x-ga.theorem.com"/>
   <attribute name="ServerPort" value="443"/>
   <attribute name="ServerRootURI" value="3dspace"/>
   <attribute name="LoginTicket"
value="MUQ3NThFQkI3NjQzNDc0NkFBRkRGMkJBN0I0MkZBNzI8c3RlcGhlbmNsZXdzfHx8fDB
8" />
   <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-48351120-00012299"/>
   <attribute key="V_version" value="A"/>
  </arguments>
 </mc6_read>
```

The PLM_ExternalID value will be the Name displayed in the properties of the object selected in 3DExperience. The V_version value will be the Revision displayed.



Please note that in some cases, a “minorrevision” value may be required in the XML file. For example, if the revision is A.1 then a value of 1 may be required. This attribute can be added below the V_version attribute in the XML file as per the format below:

<attribute key="PLM_ExternalID" value="prd-48351120-00012299"/>
<attribute key="V_version" value="A"/>
<attribute key="minorrevision" value="1"/>

-o <output_file>

Is the required JT output file name.

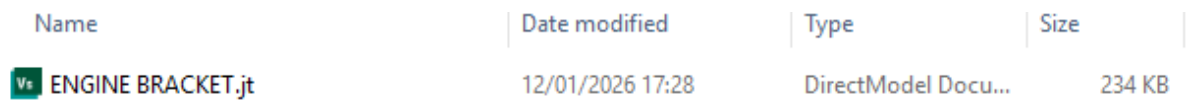
-z <config_file>

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. disable_points, disable_wireframe. ***(See Configuration Manager, 3DExperience to JT for list of command line syntaxes.)***

```
C:\Users\stephen.clews>"C:\Program Files\Theorem\28.2_3DExperience_JT\bin\catia6r2025x_jt.cmd"
C:\Temp\3dex_2025x_input_Engine_Bracket.xml -o "C:\Temp\ENGINE BRACKET.jt" -z "C:\Program Files
\Theorem\28.2_3DExperience_JT\etc\mono.config" disable_wireframe
```



Log File Generation

Export Process Log Files

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

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

The process log and error messages are, by default, located in the 3DExperience CATReport directory, e.g.

C:\Users\<user>\AppData\Local\DassaultSystemes\CATReport

Where <user> is your user name

The files are named the same as the part or assembly being exported e.g. ENGINE_BRACKET would produce the following log file names:

- ENGINE_BRACKET.err
- ENGINE_BRACKET.rpt
- ENGINE_BRACKET.log.summary
- ENGINE_BRACKET.log

Additional log files are created in the TSC_TEMP_DIR directory. This directory is defined in the ***ts_env.bat*** file which can be accessed from within the SpinFire Convert install.

%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 3DExperience 'data read' processing into the SpinFire Convert 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
```

%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. These are the same types of files produced when exporting from 3DExperience.

- .err file
- .rpt file
- .log.summary file
- .log file

The process log and error messages are, by default, located in the 3DExperience CATReport directory, e.g.

C:\Users\<user>\AppData\Local\DassaultSystemes\CATReport

Where <user> is your user name

The files are named after the selected input file name. e.g. nist_ftc_08_asme1_ct5240_rc.jt would produce the following log file names:

- nist_ftc_08_asme1_ct5240_rc.err
- nist_ftc_08_asme1_ct5240_rc.rpt
- nist_ftc_08_asme1_ct5240_rc.log.summary
- nist_ftc_08_asme1_ct5240_rc.log

3DExperience Environment Files

As part of the SpinFire Convert installation process, a set of 3DExperience environment files are created which are subsequently used in the launch of 3DExperience and CATUTIL sessions to support the plug-ins for JT import and export.

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

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

<installation_directory>\B425\win_b64\CATEnv\2023x_JT.txt

<installation_directory>\B426\win_b64\CATEnv\2024x_JT.txt

<installation_directory>\B427\win_b64\CATEnv\2025x_JT.txt

These files consist of the current 3DExperience settings along with the required SpinFire Convert settings appended at the bottom.

```
!-------------------------------------------------
! SpinFire Convert Additional Multi-CAD Settings
!-------------------------------------------------
TS_INST=C:\Program Files\Theorem\28.2_3DExperience_JT\
THEOREM_LICENSE_FILE=7601@ts-tam-lic-svr
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\28.2_3DExperience_JT\\data\jt\jt_export_name_format.txt
TS_JT_TEMPLATE_FILE=C:\Program Files\Theorem\28.2_3DExperience_JT\\data\jt\template.jt
TS_JT_TEMPLATE_FILE_EMPTY=C:\Program Files\Theorem\28.2_3DExperience_JT\\data\jt\template_empty.jt
TS_JT_TEMPLATE_FILE_HIDDEN=C:\Program Files\Theorem\28.2_3DExperience_JT\\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\28.2_3DExperience_JT\\data\jt\log_processing.txt
!
!For Interactive or DataExchangePLMBatch Large Assembly Processing, set the following
!TS_V6_LAP_INPUT_TEMPLATE=Name_of_a_Theorem_format_batch_xml_input_data_file
!For XPG usage where the original managed err log is to be used, set the following
!TS_XPG_USE_EXISTING_ERR_LOG=1
!
!To enable automatic TDP upload to the SpinFire XR server, set the following TS_TDP_ values
!TS_TDP_ServerUrl=http:\\xr_server_host_name:9000
!Leave TS_TDP_UserEmail and TS_TDP_UserPassword blank if using the Windows Credential Manager
!TS_TDP_UserEmail=User Email
!TS_TDP_UserPassword=User Password
!TS_TDP_Folder=Common or Self
!
!Enable PLM Product Reference Object Attributes query by type definition
!Modify as appropriate to your PLM Installation
TS_PLM_PRODUCT_REF_TYPE_NAME=VPMReference
TS_PLM_REPRESENTATION_TYPE_NAME=VPMRepReference
```

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 -z.

For interactive users the configuration file can be defined in the configuration manager within Preferences>App Preferences>SpinFire>SpinFire Convert

By default, this is set to TS_INST\etc**tess.config**

Alternatively, if translating using DataExchangePLMBatch, with the environment variable TS_IGNORE_JT_CATSETTINGS=1 set, 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

"%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 **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 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	*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		

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		

Below is an example JT configuration file.

```
version "EAITranslator" 1.0.1 "EAITranslator"
EAITranslator {
    OutputDirectory = "./"
    CommonPartsPath = ""
    chordalOption = "ABSOLUTE"
    structureOption = "PER_PART"
    writeWhichFiles = "ALL"
    compression = true
    advCompression = true
    advCompressionLevel = 0.0
    JtFileFormat = "95"
    triStripOpt = true
    seamSewing = false
    seamSewingTol = 0.001
    includeBrep = true
    includeGeom = true
    autoXtBrep = false
    brepPrecision = "DOUBLE"
    autoNameSanitize = false
    nameSanitizeMacro = ""
    updateChangedPartsOnly = false
    verboseReporting = false
    writeAsciiAssembly = false
    singlePartsNoAssem = true
    autoLowLODgeneration = false
    smartLODgeneration = true
    numLODs = 3
    includeULP = "PASSTHROUGH"
    ulpPrecision = 0.001
}
Filter {
}
Metadata {
}
LOD "1" {
    Level = 1
    Chordal = 0.2
    Angular = 5
    Length  = 0.0
    Label = "ud_FINE"
    FeatureSuppression = 0
    Simplify = 1.0
    AdvCompressionLevel = 0.0
    ULP = false
}
```

Appendix B – Large Assembly Processing (LAP)

Overview

The export of large assemblies from 3DExperience 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 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 3DExperience to 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)**	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.

Command Line Operation

The options for Large Assembly Processing can be used as command line options on the catia6_jt.cmd. ***(See Translating on the Command Line.)***

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.

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

e.g. <installation_directory>\B427\win_b64\CATEnv\2025x_JT.txt

TS_V6_LAP_INPUT_TEMPLATE=<XML File name>
TS_PLM_PRODUCT_REF_TYPE_NAME= VPMReference
TS_PLM_REPRESENTATION_TYPE_NAME= VPMRepReference

Where <XML File name> is the XML file used, e.g. 3dex_2022x_input_Engine_Bracket.xml. ***See Translating on the Command Line*** for more information on XML Files.

Appendix C – 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 include the TAG attribute but under a different name, i.e. PART NUMBER
TAG,PART NUMBER,1,,,

Appendix D – 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 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.*

Appendix E – JT Template Files

If, when a representation is read from 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 SpinFire Convert CATEnv file e.g. <installation_directory>\B427\win_b64\CATEnv\2025x_JT.txt, typically as follows:

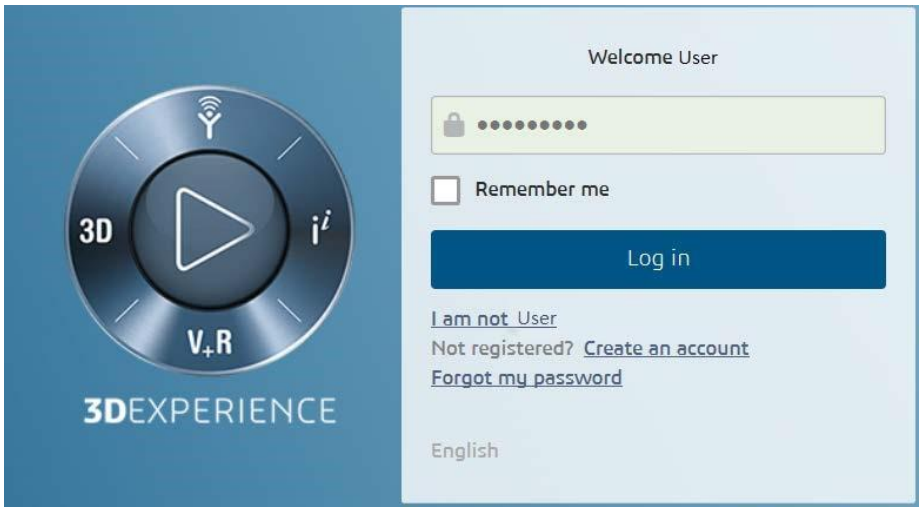
TS_JT_TEMPLATE_FILE=C:\Program Files\SpinFire\SpinFire Convert\28.2\data\jt\template.jt
TS_JT_TEMPLATE_FILE_EMPTY=C:\Program Files\ SpinFire\SpinFire Convert \28.2\data\jt\template_empty.jt
TS_JT_TEMPLATE_FILE_HIDDEN=C:\Program Files\ SpinFire\SpinFire Convert \28.2\data\jt\template_hidden.jt

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

Appendix F – Creating a Login Ticket

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

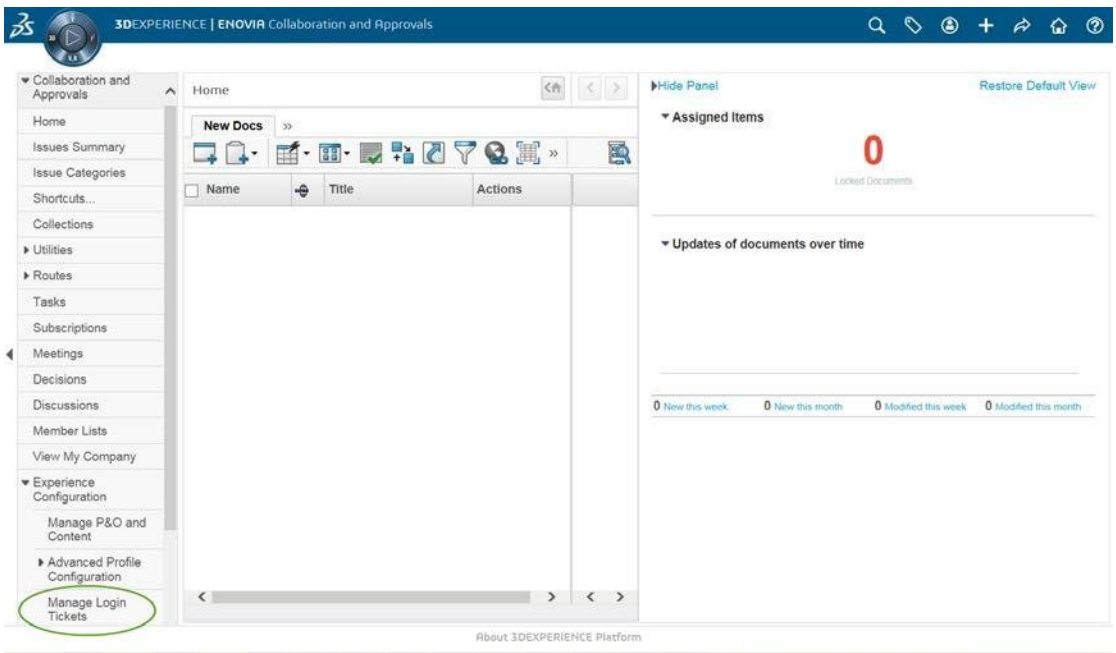
https://3dspace.2017x.theorem.com:447/3dspace/common/emxNavigator.jsp

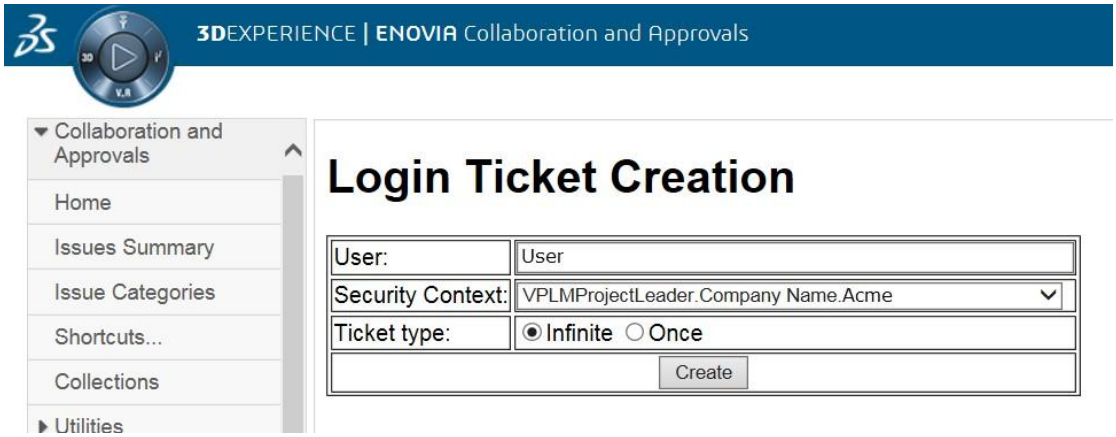


Login as normal, e.g.

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'.

This will produce the ticket. See example below: -

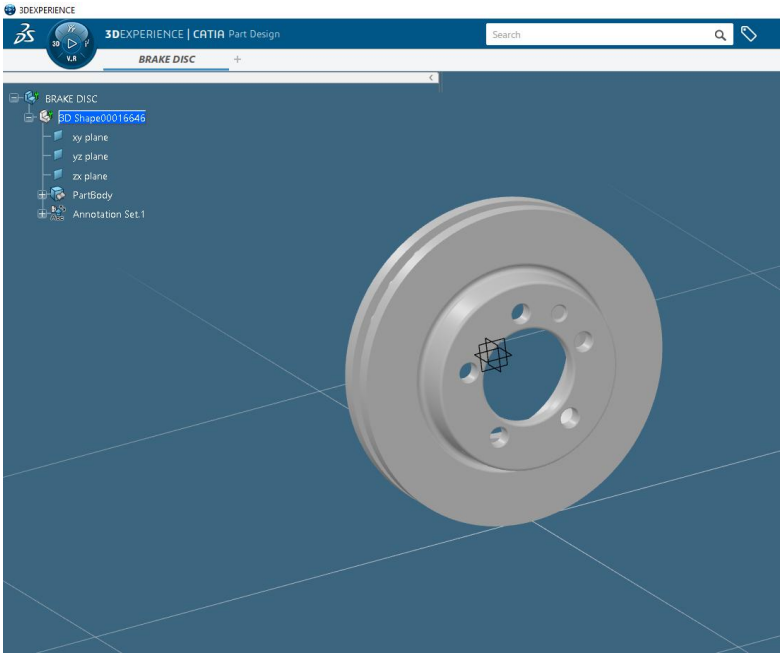
REEyNzM3STE1MER2NDgxQzlFNzk1QzIwNjZGNATzDAN8Um9iaW58Um9iaW58fHwwfA==

This can then be used as the 'LoginTicket' value explained previously.

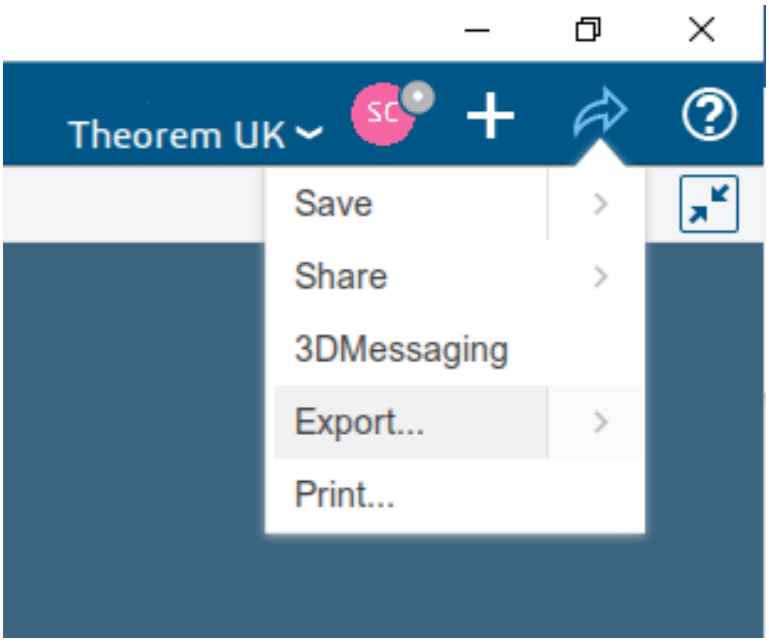
Appendix G – Translating Interactively from within 3DExperience – Version 27.0 and Earlier

Interactive Export to JT

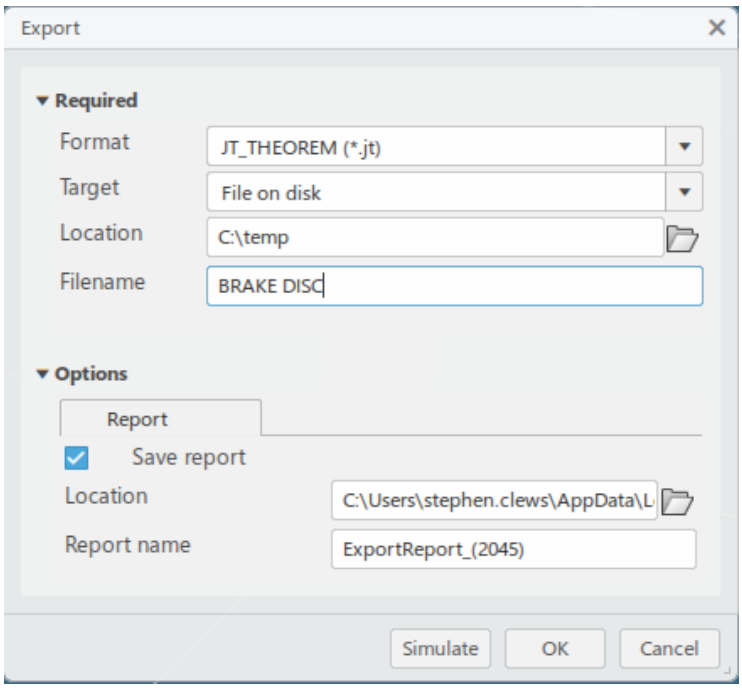
Once the 3DExperience application has been launched, open the product or representation that is going to be exported to JT.



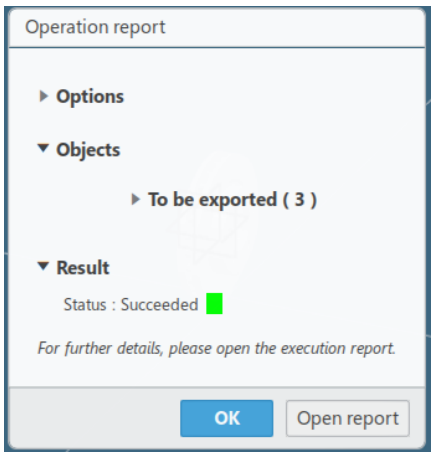
Select the arrow in the top right corner, then from the list displayed select Export.

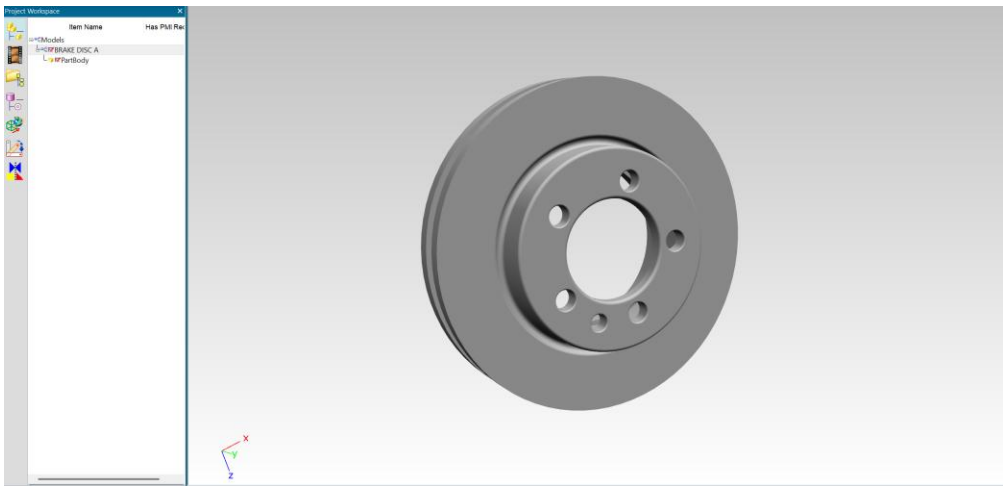


In the Export dialog box displayed, ensure the 'Format' is set to JT_THEOREM (*.jt/*.plmxml/*.stpx). Select the required location for the file and ensure the Filename displayed is correct. Click OK to initiate the export to JT.



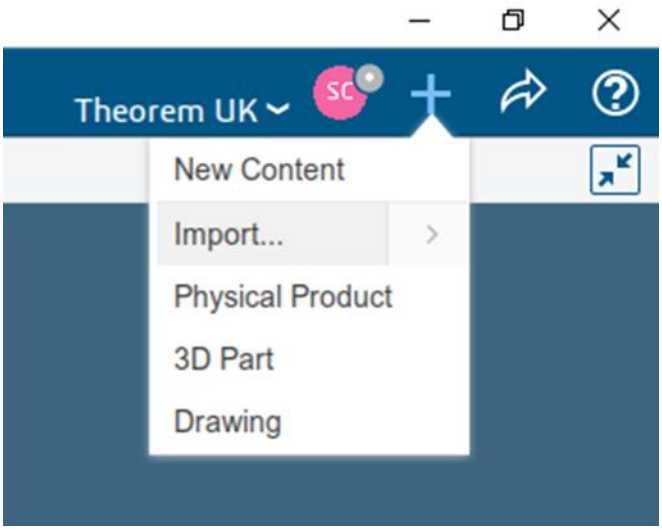
Export status of Succeeded displayed in Operation report window. JT files created in the location specified.



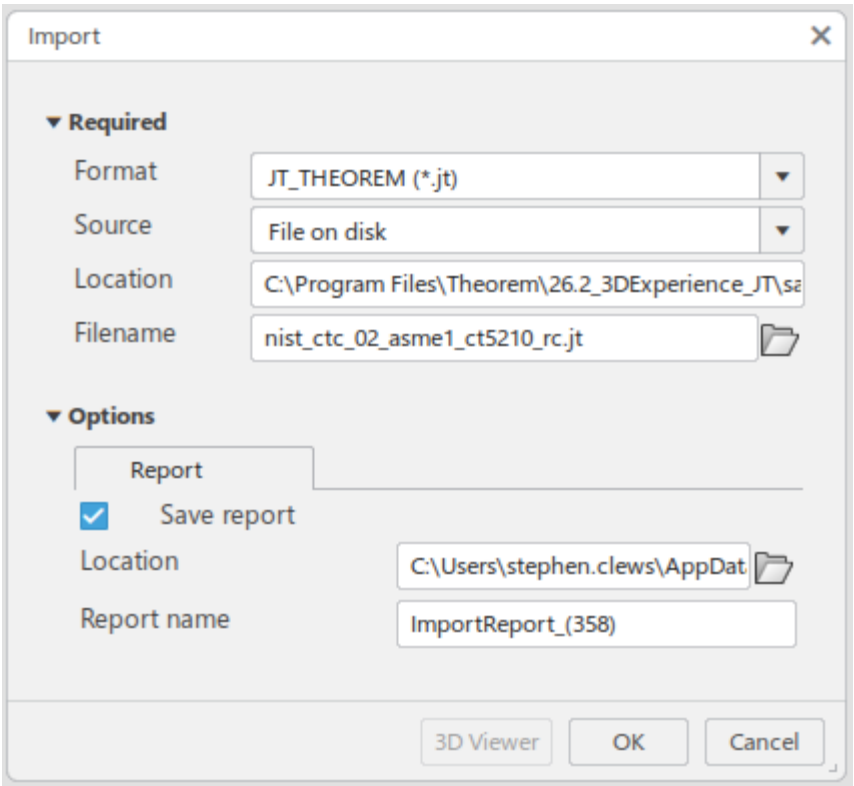


Interactive Import to 3DExperience

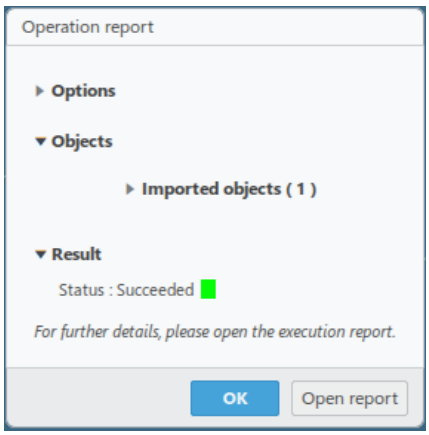
Once the 3DExperience application has been launched, it is possible for a model to be imported from JT. To do this select the **'+'** icon in the top right corner, then from the list displayed select import.

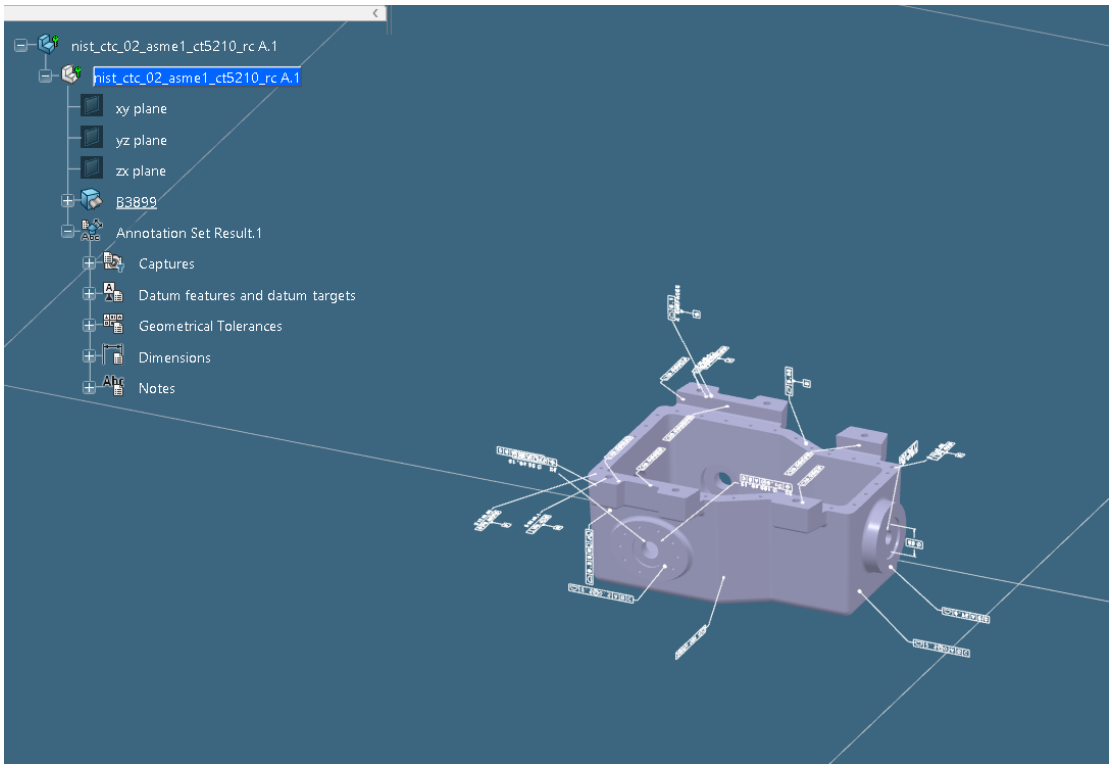


In the Import dialog box displayed, ensure the 'Format' is set to JT_THEOREM (*.jt/*.plmxml/*.stpx). Click on the folder icon next to the Filename field and choose the required JT file, then click OK to initiate the import to 3DExperience.



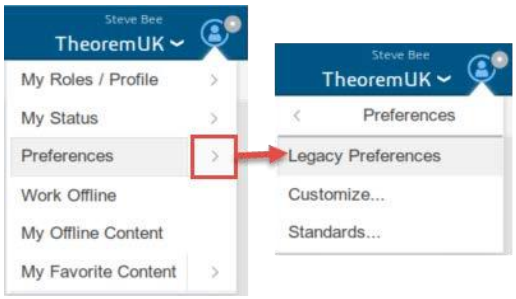
Import status of Succeeded displayed in Operation report window. Imported data saved into the 3DExperience database and opened into a new tab in the user's session.

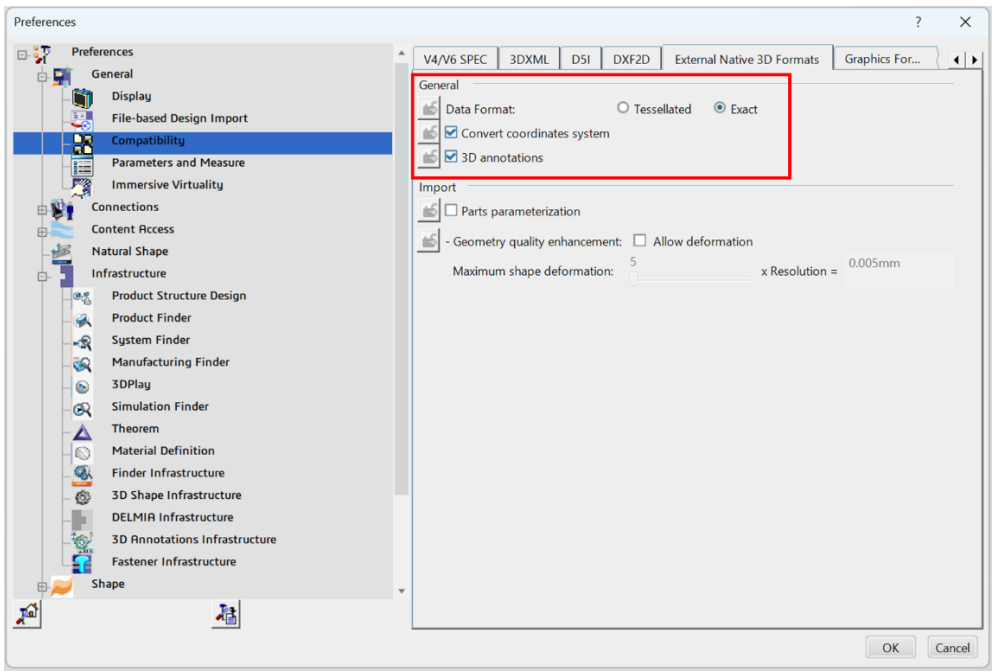




Appendix H – Theorem Interactive Conversion Settings – Version 27.0 and Earlier

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.***

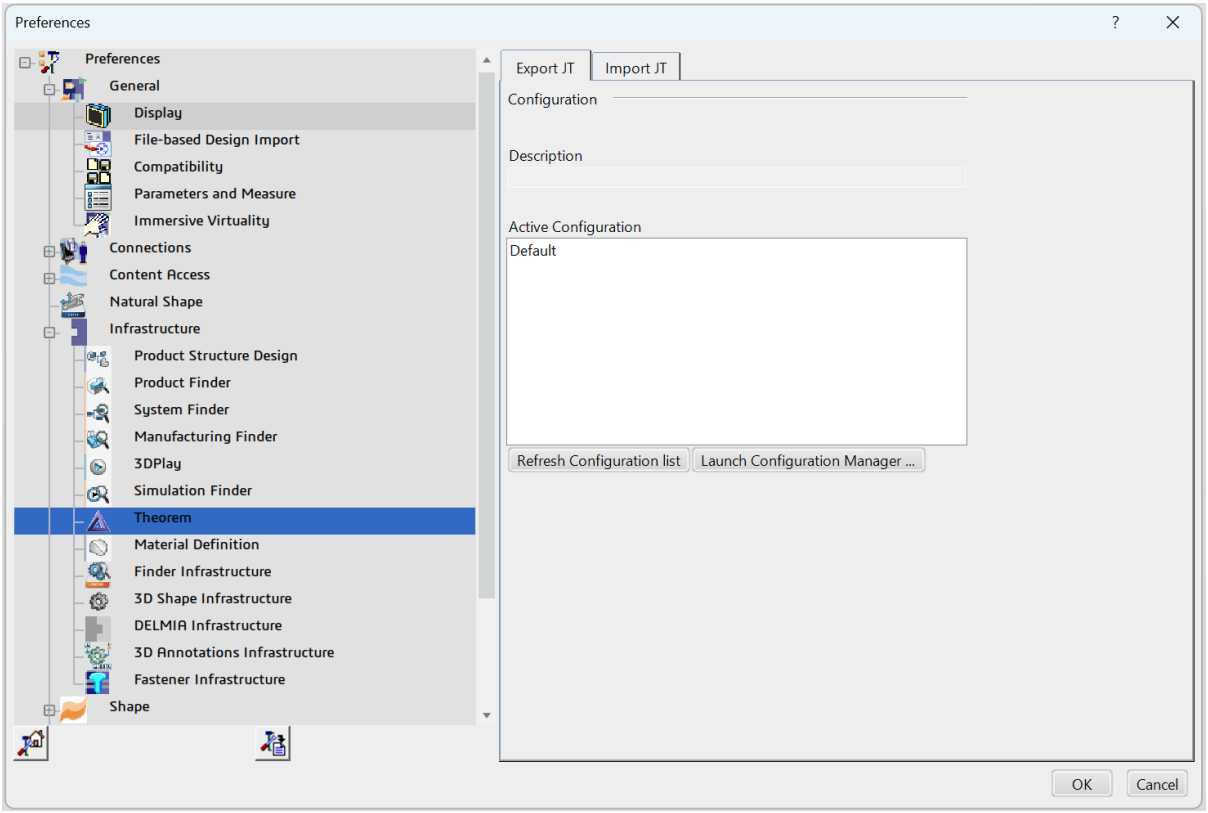




This page is a standard Dassault Page that enables the user to set the preferred mode of conversion (in this case Exact). It also enables the user to apply general options such as ***'Convert coordinates system'*** and ***'3D annotations'***.

There are also some Theorem settings that can be applied. These are accessed through ***Preferences>Legacy Preferences> Infrastructure>Theorem***

Two dedicated tabs under ***'Theorem'*** allow the user access to Theorem Configurations for importing from JT and exporting to JT. From this Panel, the user can select a predefined configuration or create a new configuration. Please follow the process defined in **Theorem Interactive Conversion Settings** for creating new configurations.



SPINFIRE
CONVERT
Powered by Theorem Technology

THEOREM HOUSE
MARSTON PARK
BONEHILL RD
TAMWORTH
B78 3HU
UNITED KINGDOM

+44(0)1827 305 350

THEOREM SOLUTIONS INC.
100 WEST BIG BEAVER
TROY
MICHIGAN
48084
USA

+(513) 576 1100

UK, Europe and Asia Pacific Regions

THEOREM HOUSE
MARSTON PARK
BONEHILL RD
TAMWORTH
B78 3HU
UNITED KINGDOM

sales.Spinfire Convert@techsoft3d.com

+44 (0) 1827 305 350

USA and the America

THEOREM SOLUTIONS INC
100 WEST BIG BEAVER
TROY
MICHIGAN
48084
USA

@techsoft3d.com

+(513) 576 1100

TECHSOFT3D.COM