User Guide
CATIA V5 - JT

Product Category	CADTranslate
Product Group	CATIA V5 - JT
Product Release Version	27.0

Document Type	User Guide
Document Status	Released
Document Revision	1.0
Document Author	Product Manager
Document Issued	31/05/2024

THEOREM.COM

Contents

Overview of CADTranslate

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:
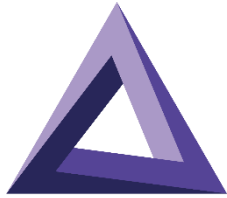
CAD**Translate**

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

See our website for more detail.

CAD**Publish**

The creation of documents enriched with 3D content

See our website for more detail.

Theorem**XR**

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

See our website for more detail.

CADTranslate for CATIA V5 - JT

The CATIA V5 to JT translator may be installed on a number of machines each accessing a central network-floating license.

The CATIA V5-JT Translator is a bi-directional direct database converter between the Dassault Systèmes CATIA V5 Modelling Application and the JT file format, used by the Siemens Teamcenter Visualization products.

It enables the user to convert all forms of 3D Mechanical Design Geometry and Assembly data, together with system defined attribute information, colour information, between these two systems. This product is designed for companies using CATIA V5 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 therefore companies using JT based solutions need to translate their CATIA V5 data into the JT format.

The translator can be invoked in either an interactive or batch mode with the command line interface allowing the conversion process to be integrated into any process oriented operation. Alternatively from Theorem R18 the conversion process may be operated by using the new Theorem Unified Interface.

Primary Product Features

- Converts all types of geometry, wire frame, surfaces, trimmed surfaces (faces) and solid models.

- Converts assembly structure between both systems.

- Converts attribute data including colour and layer information.

- Integrated with the CATIA V5 installation.

- The conversion process can be run Interactively, Batch Mode or using the new Unified Interface

- Command line interface allows process integration.

- Data can be filtered by layer and entity type during processing. Geometry can be filtered and selectively processed.

- Uses the CATIA V5 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 JT Brep definition or an XT Brep definition. As standard a facetted representation is created with the user selecting whether JT or XT Brep definition is created. Converts all types of geometry, wire frame, surfaces, trimmed surfaces (faces) and solid models.

Primary Product benefits?

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

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

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

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

This document will focus specifically on guidance for the use of the Visualize 3D for CATIA V5 – JT 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. 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 and select UI from the product selection list.

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

Using the Product

Default Translations

Default Translation – via the Unified Interface

The Unified Interface can be started via the Start Menu – if a shortcut was added during installation.

Alternatively, the Unified Interface can be run via a Windows Explorer selection in:
<UI_installation_directory>\bin\Unified_Interface.cmd

The following interface will be launched:
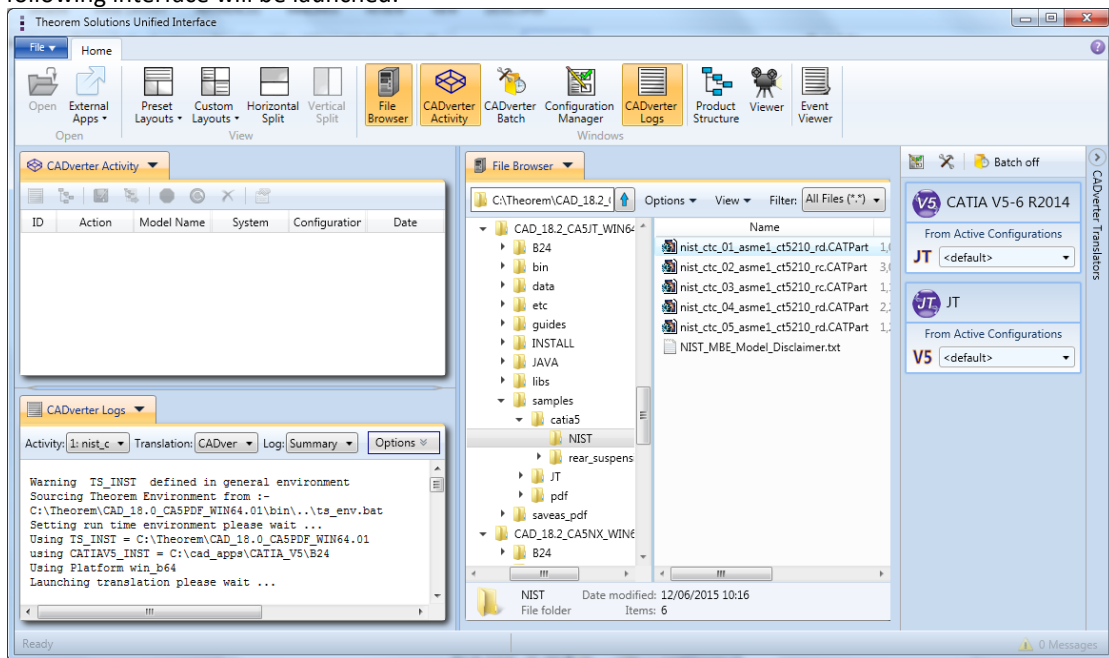
The default layout is split into 4 primary areas, which can be altered to the users prefer:
The simplest way to translate from CATIA V5 or JT is to drag a file from the file Browser Pane on to the Active Configurations for the translation you require.
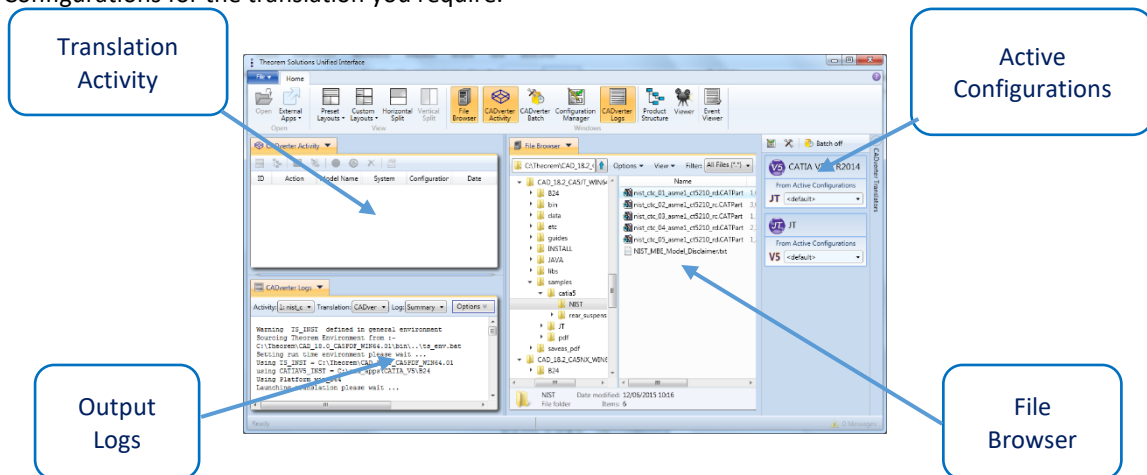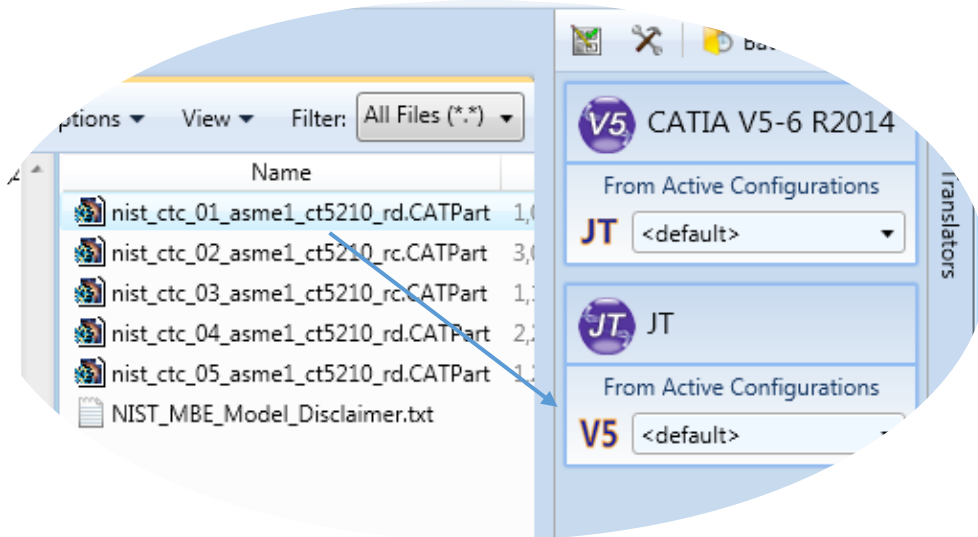
On completion, the Unified Interface will display the activity information and details from the log file created during the translation, if requested, in the Translation Activity and Output Log panes, respectively.

The generated output data can be located by selecting the translation from the Activity pane and opening the output folder:
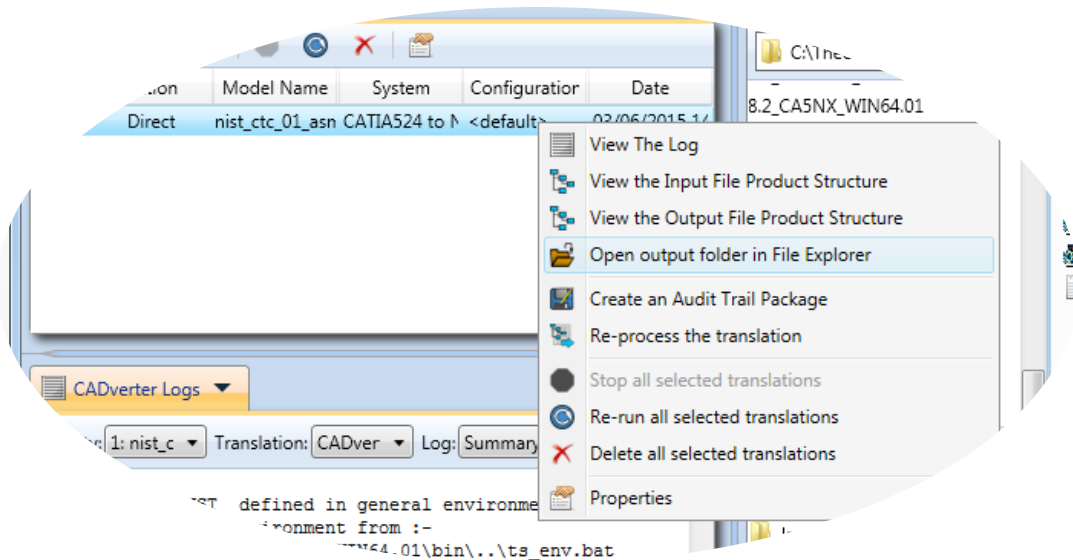
Default Translation – via the Command Line

Running a translation via the command line can be carried out via the **cad_run.cmd** file located in the **<installation_directory>\bin** directory. The format of the command is as follows when translating from CATIA V5 to JT:

    <Translator_installation_directory>\bin\cad_run.cmd CATIA5[XX]_JT –i <input_file> -o <output_file>

The format of the command is as follows when translating from JT to CATIA V5:

    <Translator_installation_directory>\bin\cad_run.cmd JT_CATIA5[XX] –i <input_file> -o <output_file>

(Note! Replace the [XX] seen in the example with the version of CATIA V5 you are using. E.g. for CATIA V5 R34 (V5-6 R2024), change to CATIA534):

```
C:\>"C:\Program Files\Theorem\27.0\bin\cad_run.cmd" CATIA534_JT –i "C:\Program Files\Theorem\27.0\samples\catia5\NIST\ni
st_ctc_01_asme1_ct5210_rd.CATPart" –o C:\TEMP\NIST_Output.jt
```

The example above will translate a CATIA V5 sample file provided within the installation and finish with the following screen output when successful:

```
Translation Complete...
      Exporting Parts and Assembly to JT Files

   ********************************
   * JT file successfully created *
   * C:\TEMP\NIST_Output.jt       *
   ********************************

   ****************************************************
   *                                                  *
   * CATIA5 to JT conversion ...                      *
   *                     ... Completed Successfully   *
   * Output JT file created                           *
   *                                                  *
   ****************************************************
```

The file will be output to the target location. In this case:

_C:\Temp\NIST_Output.jt_

Translator Customization

The Theorem translator allows the information that is read from the source system and written to the target system to be tailored via a set of user specified arguments. Commonly used arguments are supported via the Unified Interface, with Advanced Arguments being described within this document for use in the Unified Interface or via the Command Line invocation.

Common Options for CATIA V5 to JT

Within the Configuration Manager pane of the Unified Interface, arguments that can be specified when publishing CATIA V5 data into JT are grouped into 4 areas:

- CATIA V5 Read – Those arguments that affect how data is read from CATIA V5

- JT Write – Those arguments that affect how the data is written to JT

- Entity Mask – Those arguments that allow specific read entities to be masked

- General – Those arguments that are common to ALL Publishing activities regardless of source data.
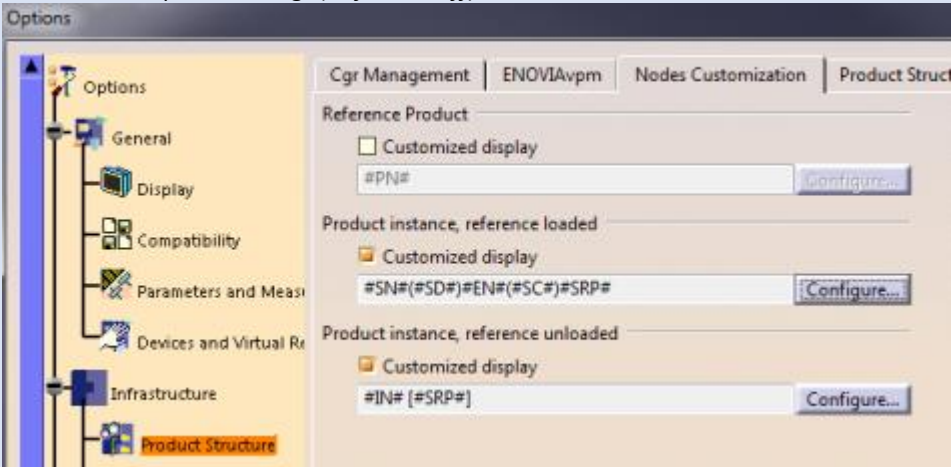
CATIA V5 Read Arguments

The image below shows the CATIA V5 Read arguments that are available, with their default settings:

CATIA V5 Read	JT Write	Entity Mask	General
Option Name		**Value**	
Read PMI		☐	
PMI Level		All	
Read Captures		☑	
Read FTA Reference Geometry		☐	
Maintain CATIA V5 Instance Names		☐	
Read Face Colours		☐	
Read Face Opacity		☐	

Each of these options is described below:

Option	Description

Read PMI	Enables PMI data read from the V5 file. *(Default is ON).*
	
	    o Command Line Syntax:
	
	          ▪ dont_read_pmi – to turn off
	
	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'*

PMI Level	A secondary argument to 'Read PMI' and allows control of the level of PMI to be read. Default is ALL when 'Read PMI' is marked as ON.
	
	• Options Available (command line syntax in italics and square brackets next to the option)
	   o All - [read_pmi]
	   o Part Level - [read_part_pmi]
	   o Assembly Level - *[read_assy_pmi]*
	   o Assembly Set (From CATPart) - *[read_part_assy_pmi]*
	   o Assembly Set (All) - *[read_all_assy_pmi]*

Read Captures	A secondary argument to 'Read PMI' and allows the control over whether captures are read as part of the process. Default is ON when 'Read PMI' is marked as ON.
	  o Command Line Syntax:
	    ▪ read_captures
	    ▪ dont_read_captures – to turn off

Read FTA Reference Geometry	Enables reading of FTA Reference Geometry *(Default is Off)*
	  o Command Line Syntax:
	    ▪ read_geometry – to turn on

Maintain CATIA V5 Instance Names	Honours CATIA V5 *Tools->Options->Infrastructure->Product Structure->Nodes Customization* panel settings *(Default is Off)*
	
	  o Command Line Syntax:
	    ▪ ditto_naming V5 – to turn on

Read Face Colours	Read the face colours (Default is OFF)
	  o Command Line Syntax:
	    ▪ read_face_colours – to turn on

Read Face Opacity	Read the opacity of the face (Default is OFF)
	  o Command Line Syntax:
	    ▪ read_materials – to turn on

JT Write Arguments

The image below shows the Write JT arguments that are available, with their default settings:

Description:	Default Translation Settings	
CATIA V5 Read	JT Write	Entity Mask	General
Option Name		Value
Config File		
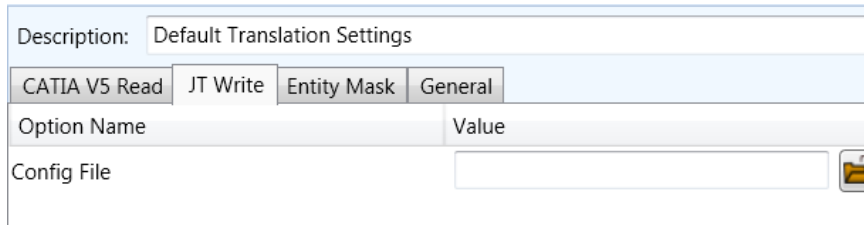
Each of these options is described below:

Option	Description
Config File	Allows a JT configuration file to be specified. Please see Appendix B for a full description of the JT config file format. o Command Line Syntax ▪ -z [path_to_file]

CATIA V5 to JT Entity Masking Arguments

The image below shows the Masking arguments that are available, with their default settings:

Description:	Default Translation Settings	
CATIA V5 Read	JT Write	Entity Mask	General
Option Name		Value
Mask File		
Entity Types Translated		
Layers Translated		
Convert NO SHOW Geometry	☐	
Convert NO SHOW Structure	☐	
Convert NO SHOW PMI	☐	

Each of these options is described below:

Option	Description
Mask File	Specifies the Mask File to be written to, that can be referenced by future translations. A Mask file MUST be specified if masking is required. The first line in this file is OFF ALL ENT: o Command Line Syntax: ▪ Mask <filename>
Entity Types Translated	Specifies a selection list from which to select which entity types are to be processed. The following types are available: "POI","CUR",”SKI",”SOL",”ISO”,"TEX","AXI"

	o Command Line Syntax: ▪ Add any of the above to the specified mask file, one entry per line prefixed by the word ON, e.g. ON POI to ensure they are considered in the translation
Layers Translated	Specifies a selection list from which to select which layers are to be processed. o Command Line Syntax: ▪ A single entry of **ON ALL LAY** must precede any Layer Mask command. ▪ Add a list or range of numbers representing layer to be processed to the specified mask file to ensure they are NOT considered in the translation e.g. OFF LAY 114,149,166,167,168
Convert NO SHOW Geometry	Enables Hidden geometry to be processed *(Default is Off)* o Command Line Syntax: ▪ Add the following entry to the Mask file ***ON NOSHOW***
Convert NO SHOW Structure	Enables Hidden Assembly Structure to be processed *(Default is Off)* o Command Line Syntax: ▪ Add the following entry to the Mask file ***ON NOSHOW STR***
Convert NO SHOW PMI	Enables Hidden PMI to be processed *(Default = Off)* o Command Line Syntax: ▪ Add the following entry to the Mask file ***ON NOSHOW PMI***

CATIA V5 to JT General Arguments

CATIA V5 Read	JT Write	Entity Mask	General
Option Name			**Value**
Mass Properties			☐
Retain Assembly Structure			☑
Advanced			

The image below shows the General arguments that are available, with their default settings.

Each of these options is described below:

Option	Description
Mass Properties	Allows Mass Property information to be read from the source data and written as attributes *(Default is OFF)*

	o Command Line Syntax: ▪ mprops
Retain Assembly Structure	Maintain the assembly structure during translation. *(Default is ON)* o Command Line Syntax: ▪ offditto – to turn off
Advanced	Allows any of the Command Line Advanced arguments documented to be passed to the Unified Interface invocation.

Processing CATIA V5 FTA data to JT PMI

When using the optional CATIA V5 to JT PMI add-on module all CATIA V5 FTA data can be translated to JT PMI output. This includes the translation of 3D dimensions and annotations which are mapped to the equivalent JT PMI entities. In addition, CATIA V5 3D sections are also able to be translated. The CATIA V5 Capture definitions, which manage the visibility of selective FTA elements as well as the view zoom and orientation, are directly mapped to JT Model Views.

Options for Processing FTA Data with Filled Text

When creating 3D dimensions and annotations with the CATIA V5 FTA module the user can create the text using either stick or filled text fonts. By default, the processing of CATIA V5 filled fonts will recreate the JT output using JT text glyphs. One benefit of using JT glyph-based text definitions is that when the model is reoriented, such that the initial view of the text is reversed, the user can select the annotations to be automatically reoriented. If the filled font representation was created with a simple polygon fill only the character outline would be reoriented and the polygon fill would remain in its original location making the output unreadable.
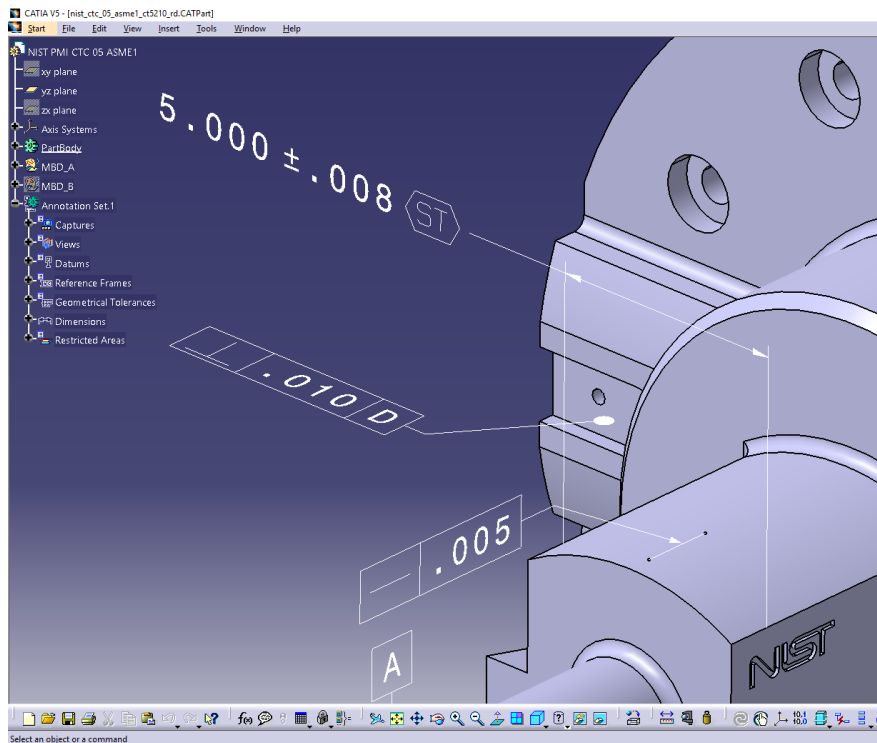
Picture showing typical CATIA V5 3D Dimensions and Annotations. Note, in the upper dimension the dimensions values 5.000 +- .008 are created using a filled text font. The characters "ST" in the frame are created using an unfilled stick font.
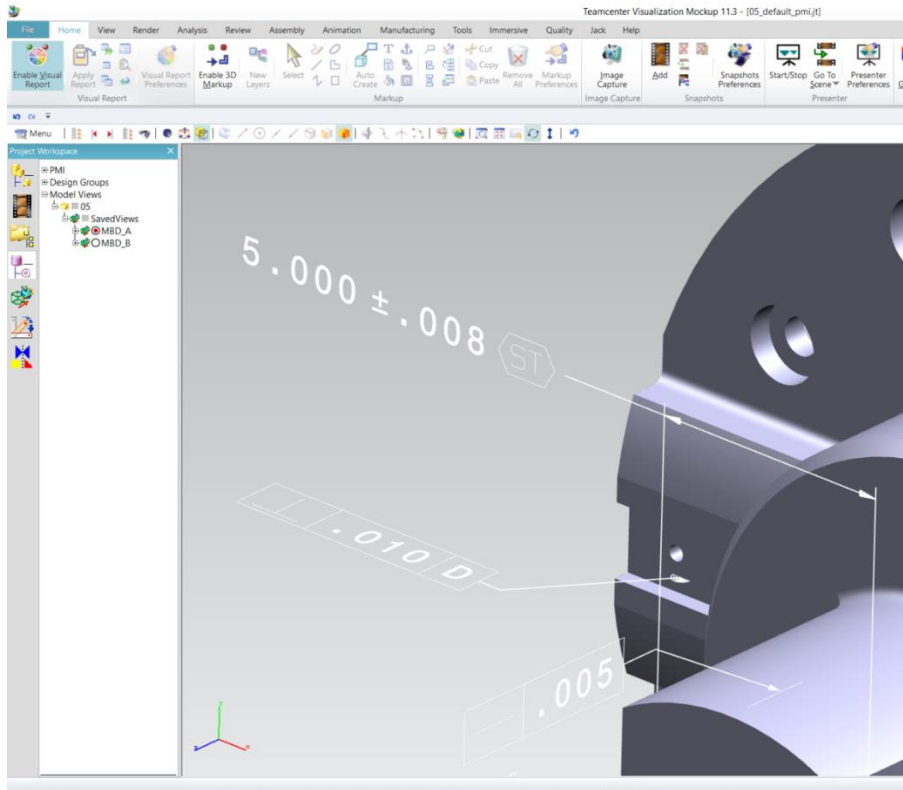
Picture showing default JT output of V5 FTA data mapped to JT PMI representation
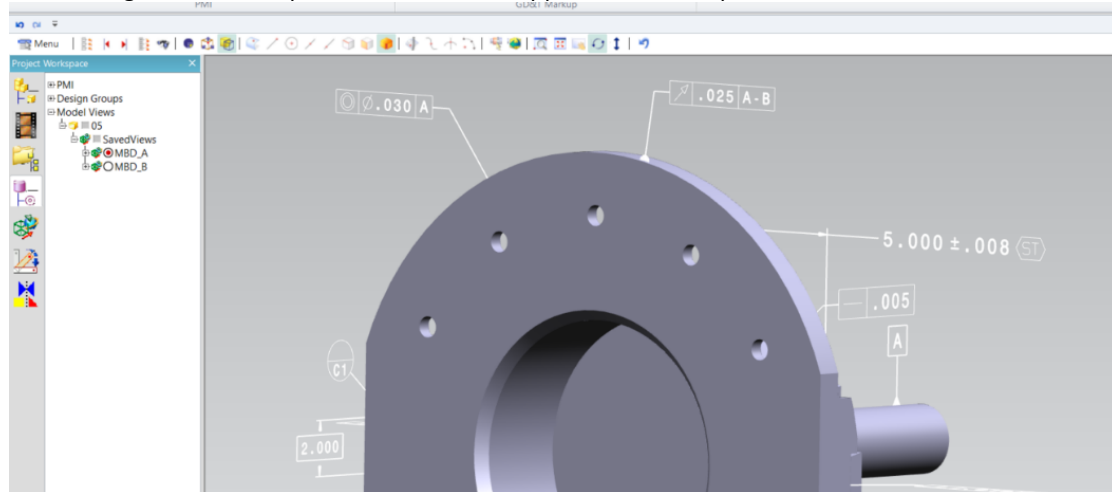
Picture showing the reorientation of the model with the text also reoriented. Note the text characters have been rotated 180 degrees and are now readable with filled character representation.

To recreate the JT filled text using simple polygons, rather than JT text glyph representations, use the options **pmi_glyphs_off pmi_polygons** together. The output representation will appear filled and look correct when viewed in the authored orientation. However, filled text created with simple polygons, will not display correctly if the data is reoriented and the text switching in JT is selected. The picture below shows an example of the display when the data is rotated, and the text is reoriented. As you will see the simple polygon fill is still in its original orientation making the dimension unreadable.

Options for Processing FTA Data using Outline Text

When processing V5 FTA text data created with filled font definition the user can select to generate the JT text representation as outline text, therefore removing the internal rendering. This is achieved by using the option **pmi_glyphs_off**.
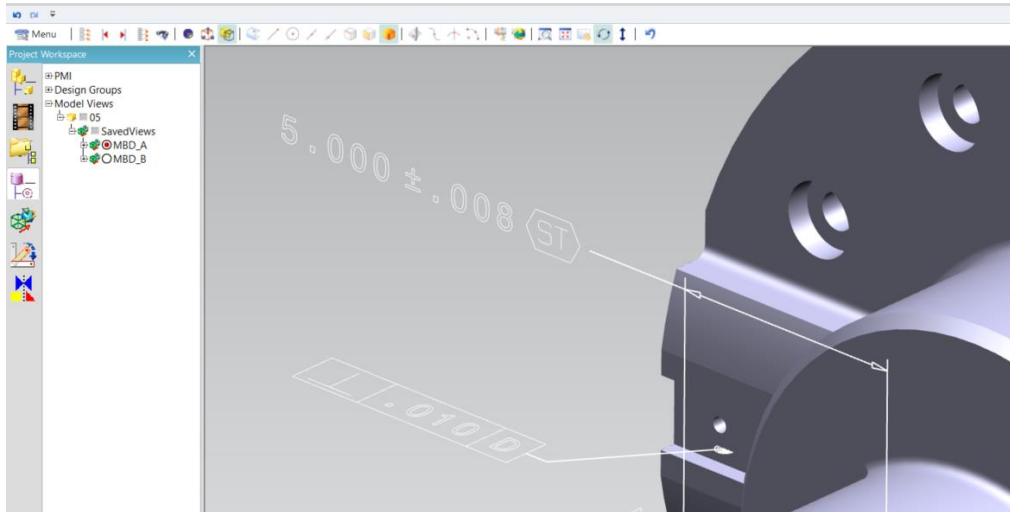
Picture showing JT PMI with filled text rendered in outline only. Note the option pmi_glyphs_off also removes the filled arrowhead rendering.

If you would prefer to maintain filled arrowheads and only impact the characters to be recreated in outline mode, then use the option **dont_fill_pmi_text** instead of the option **pmi_glyphs_off**. The resultant output is shown in the picture below with the text drawn in outline and the arrowheads filled.

THEOREM
SOLUTIONS
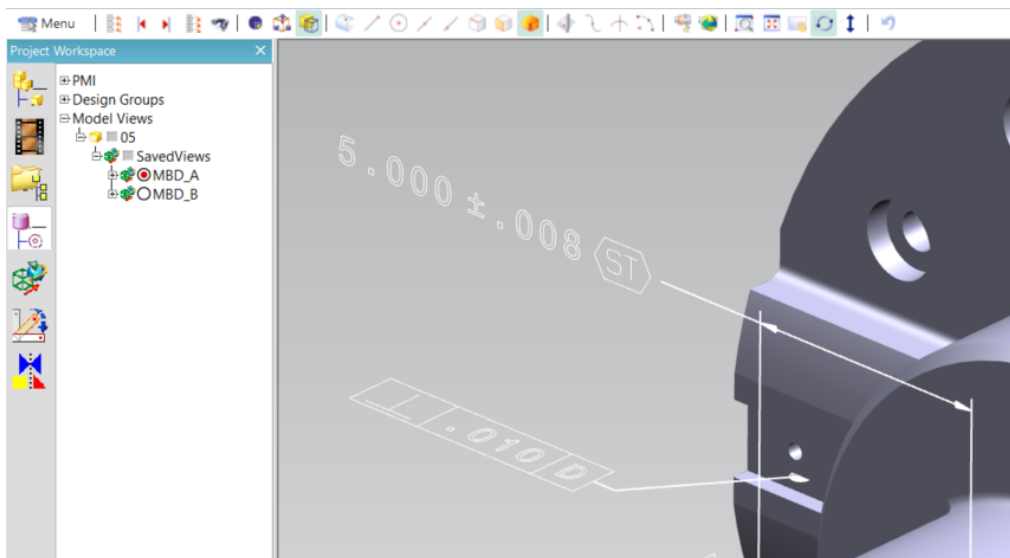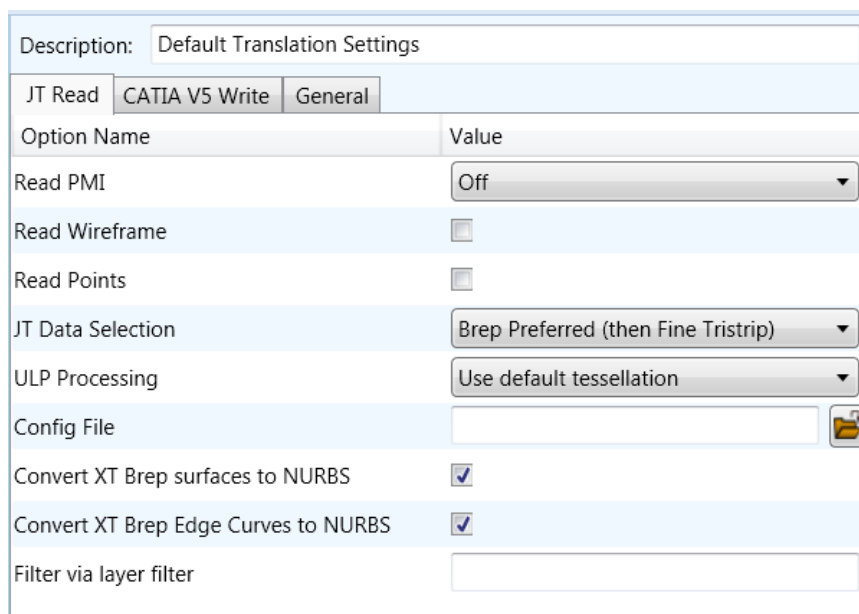
Common Options for JT to CATIA V5

Within the Configuration Manager pane of the Unified Interface, arguments that can be specified when publishing JT data into CATIA V5 are grouped into 4 areas:

- JT Read – Those arguments that affect how data is read from JT

- CATIA V5 Write – Those arguments that affect how the data is written to JT

- General – Those arguments that are common to ALL Publishing activities

    regardless of source data

JT Read Arguments

The image below shows the JT Read arguments that are available, with their default settings:

Each of these options is described below.

Option	Description

Read PMI	Reads 3D PMI. Default is OFF.
	
	    o Command Line Syntax
	
	        ▪ read_pmi dim2_pmi

Read Wireframe	Read JT wireframe data. Default is OFF.
	
	    o Command Line Syntax
	
	        ▪ read_wire_frame

Read Points	Read JT point data. Default is OFF.
	
	    o Command Line Syntax
	
	        ▪ read_points

JT Data Selection	Select Brep or tessellated data read. Default is 'Brep Preferred (then fine facet)'. o Command Line Syntax

	▪ Brep Preferred (then Fine Facet): brep_pref
	▪ Brep preferred (then Fine Tristrip): brep_pref_tri
	▪ Brep Only: brep_only
	▪ Fine Tristrip: fine_tristrips
	▪ Coarse Tristrip: coarse_tristrips
	▪ Fine Facet: fine_facets
	▪ Coarse Facet: coarse_facets

ULP Processing	How to process ULP data in the JT part. Default is 'Use Default Tessellation'.
	
	    o Command Line Syntax
	
	       ▪ Use Default Tessellation: Default
	
	       ▪ Tessellate ULP: tess_ulp

Config File	Config File for Brep or ULP tessellation.
	
	    o Command Line Syntax
	
	       ▪ -z [path to file]

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

Convert XT Brep Edge Curves to NURBS	Read XT Brep edge curves as NURBS curves (else read in native form. Default is ON.
	
	    o Command Line Syntax
	
	       ▪ rd_native_edge – to turn off

Filter via layer filter	Supply layer filter(s) separated by commas and double quoted. Default is OFF.
	
	    o Command Line Syntax
	
	       ▪ layer_filter

CATIA V5 Write Arguments

The image below shows the CATIA V5 Write arguments that are available, with their default settings:

JT Read	CATIA V5 Write	General
Option Name		Value
Output Geometry File Type		CATPart ▾
Write Face Colours		☐
Show Reference Planes		☐
Property Mapping File		[ ] 📂

Each of these options is described below:

Option	Description
Output Geometry File Type	Output Geometry file type. Default is CATPart. ○ Command Line Syntax ▪ CATPart: output_type CATPart ▪ Model: output_type model ▪ Cgr: output_type cgr ▪ Igs: output_type igs ▪ CATShape: output_type CATShape ▪ Tessellated: create_cgr
Write Face Colours	Writes face colours. Default is ON. ○ Command Line Syntax ▪ FACE_COLOUR ▪ SOLID_COLOUR – to turn off
Show Reference Planes	Creates reference planes. Default is OFF. ○ Command Line Syntax ▪ Show: dont_blank_planes ▪ No Show: Default
Property Mapping File	Path to Property Mapping File ○ Command Line Syntax ▪ cad_prop_map_file <file>

JT to CATIA V5 General Arguments

JT Read	CATIA V5 Write	General	
Option Name		Value	
Retain Assembly Structure		☑	
Advanced			

The image below shows the General arguments that are available, with their default settings:

The option is described below:

Option	Description

Retain Assembly Structure	Maintain the assembly structure during translation. *(Default is ON)*
	
	  o Command Line Syntax:
	
	    ■ offditto – to turn off

Advanced	Allows any of the Command Line Advanced arguments documented below to be passed to the Unified Interface invocation

Command Line Advanced Arguments

Advanced arguments can be added to the Command Line or the UI General->Advanced field.

CATIA V5 Advanced Arguments

Argument	Description
convert_curves	Converts curves to NURBS form
convert_surfaces	Converts surfaces to NURBS form
dont_fill_pmi_arrows	Disables the read of filled arrow information
dont_fill_pmi_text	Disables the read of filled text information
dont_read_captures	Disables the read of PMI Capture information
face_opacity	Read face opacity
no_face_colour	Sets the default to SOLID colours
noshow noshow_geom noshow_struct noshow_pmi	Reads hidden geometry / structure / pmi
output_mbd	Allows sub-part specification tree information to be read and presented to JT as product structure information.
read_geometry_edges	CATIA V5 allows the display of shaded surfaces and edges. This option allows the translator to mimic this for FTA construction geometry by promoting the edge curves to standalone wireframe.
cad_prop_map_file <file>	Creates a cad property mapping file containing all attributes if one doesn't exist.
off_ditto	Removes all assembly structure and collapses ALL geometry into a single selectable object

JT Advanced Arguments

Argument	Description
read_wire_frame	Enables reading of Wire Frame entities
read_ref_planes	Reads reference planes

Translating Interactively from within CATIA V5

The CATIA V5 to JT translator allows an active CATIA V5 Part or Assembly to be translated directly into JT or a JT Part or assembly to be imported, directly from the CATIA V5 application.
In order to translate from within CATIA V5, the CATIA V5 application must be started from within a Theorem environment, so that the appropriate CATIA V5 menus are loaded.

CATIA V5 can be started from a shortcut, if requested at installation time. Alternatively, it can be started via the script provided in the Translator installation at:

<installation_directory>\bin\catia5r[version]_start.cmd

(where [version] should be substituted for the version of CATIA V5 that you have installed – e.g. 28, 29, 30, 31):
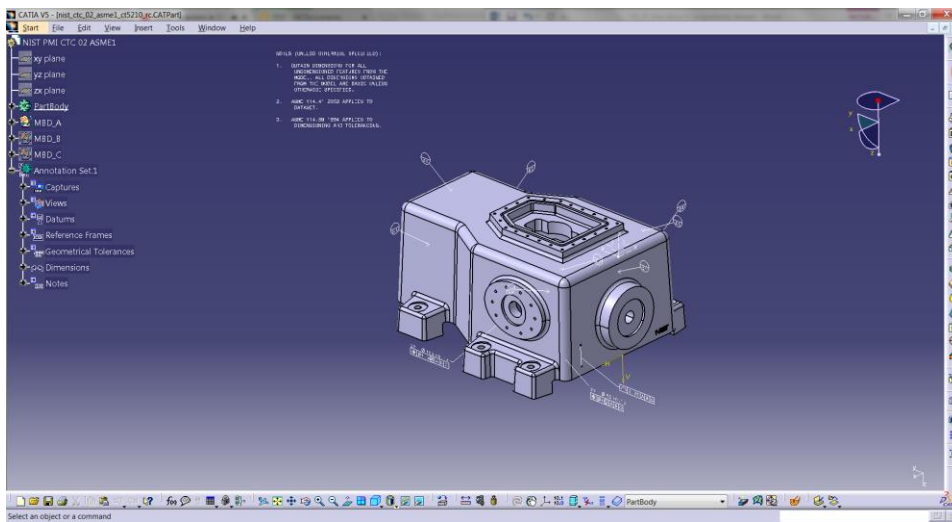
Save As JT

Once CATIA V5 has been started it is possible for a model to be saved to JT. To do this, the user selects the *File -> Save As* Menu Option. From the file browser window that opens, change the *'Save as type'* option to be **jt (*.jt)**.
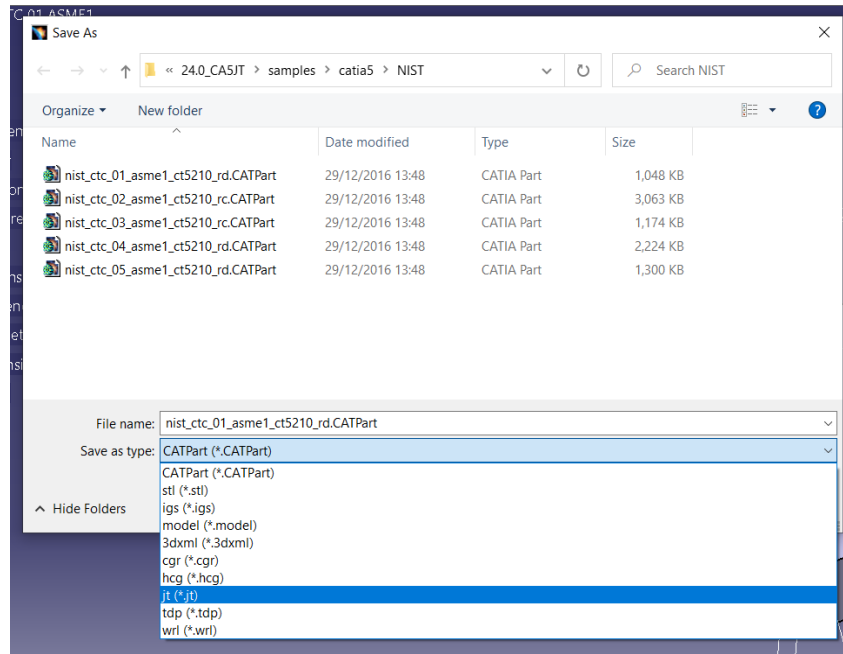
Once selected, press the 'Save' button to create the JT data.

To add a user defined configuration to the process (created from the Theorem UI), this can be selected from the **Tools>Options>Infrastructure>Theorem>CAA Export JT** menu. The process will then use this defined configuration to create the JT data as required.
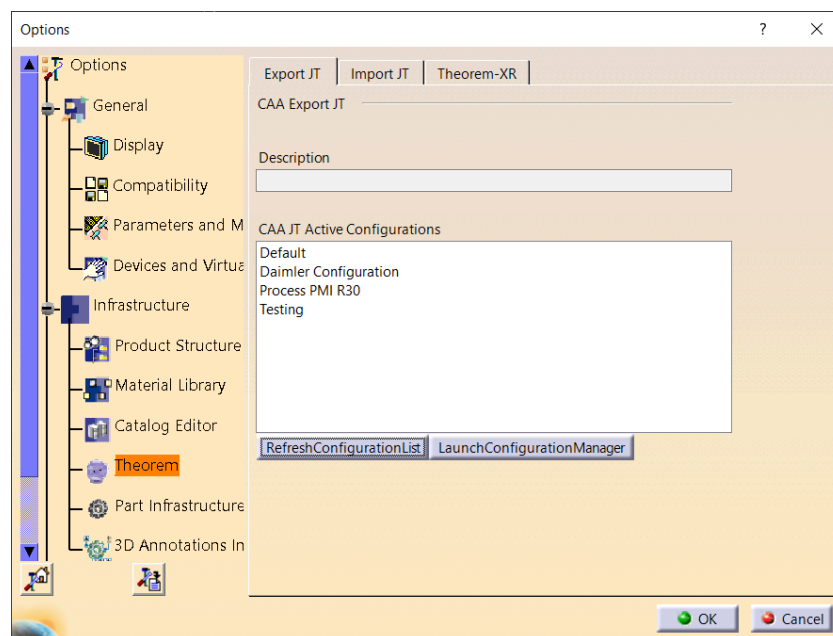
Open JT Data

Once CATIA V5 has been started it is possible for a model to be imported from JT using the *File -> Open* menu. Once selected, find the JT data you wish to import into CATIA V5 and select it from the file browser. When ready, press the 'Open' button from the menu to import.
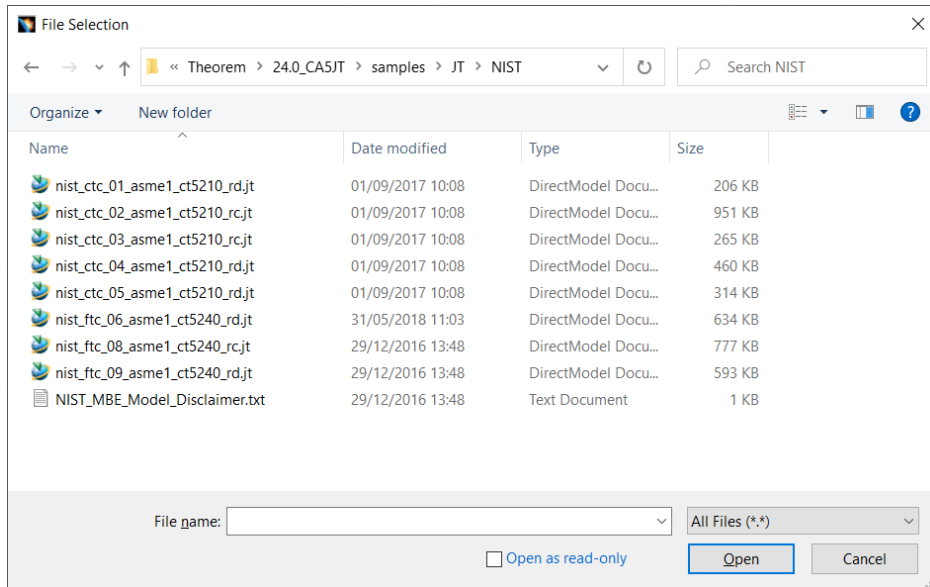
To add a user defined configuration to the process (created from the Theorem UI), this can be selected from the **Tools>Options>Infrastructure>Theorem>CAA Import JT** menu. The process will then use this defined configuration to create the JT data as required.
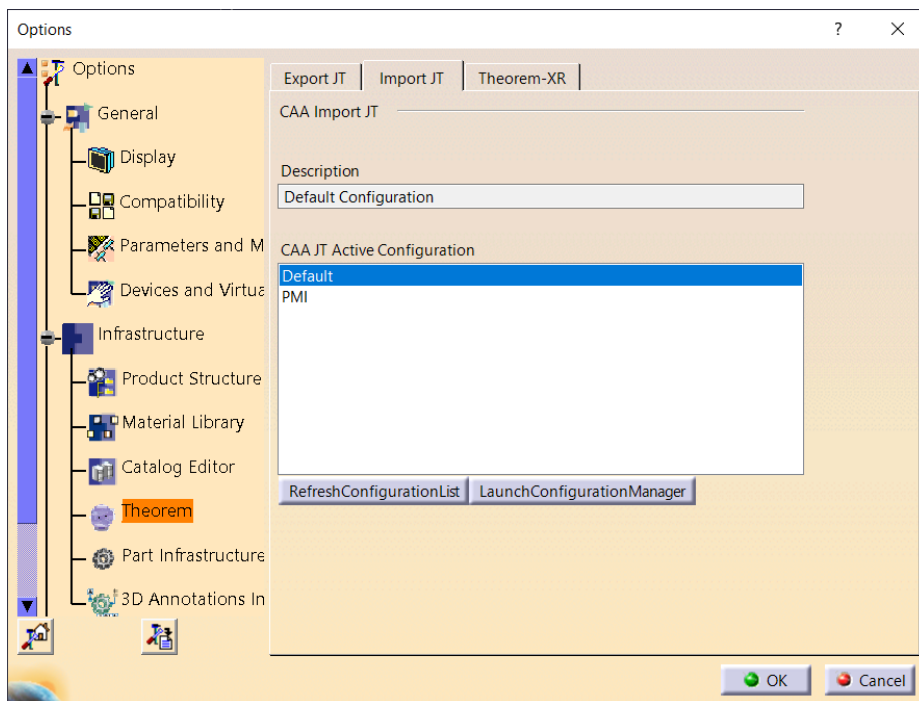
Theorem Import and Export

From the 'File' menu in CATIA V5 there are two options named 'Theorem Import' and 'Theorem Export'. This allows you to import JT data or export to JT from the File menu. Select 'File' > 'Theorem Export' or 'Theorem Import' to open the menu.
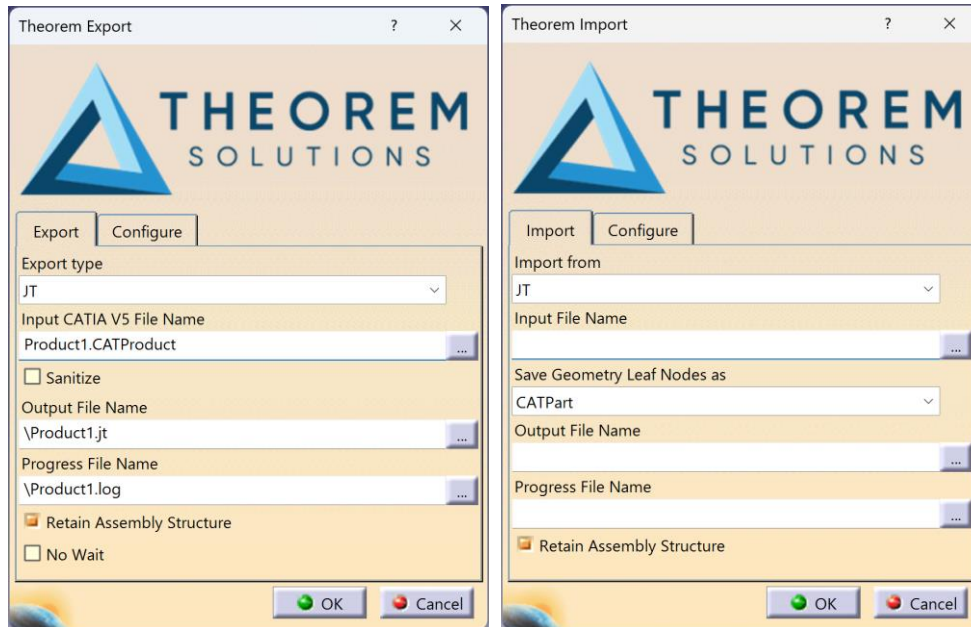
Provide output file name and log file location, then press 'OK' to export the data.

To add a custom configuration, select the 'Configure' tab.
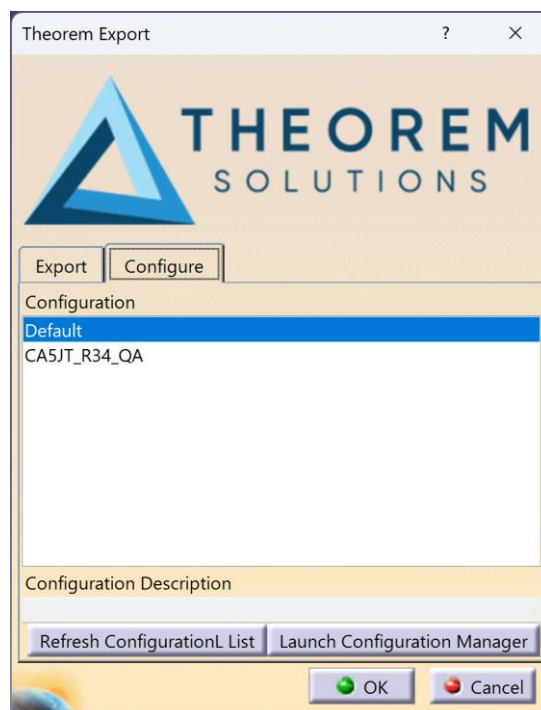
This allows a user to select their configuration, or create new ones by selecting the 'Launch Configuration Manager' option. Once a configuration is decided simply select it from the list for it to be applied.

Appendix A – CATIA V5 Configuration

Introduction

This Appendix details how to define and configure the CATIA V5 and Theorem environment to work together.

Conventions

Release of CATIA V5
To indicate a release of CATIA V5 the notation <XX> shall be used. This needs to be replaced with the specific release to be used i.e. 28, 29, 30, 31.

Platform specific directory
Within the installation directory of CATIA V5 there is a platform specific directory i.e. win_b64. This directory shall be referred to as ***<OSDS>*** in this Appendix.

Theorem Installation directory
The Theorem translator installation directory is set at installation time in the translator ***ts_env.bat*** file. This directory shall be noted as ***<%TS_INST%>*** in this Appendix.

CATIA V5 Installation Directory

Upon installation of a CATIA V5 product the user will be asked to specify the installation directory. This is the directory which contains the platform specific <OSDS> directory.

Having selected the CATIA V5 installation directory via the browse button, the installation process will record the location of the CATIA V5 installation directory in the ts_env.bat file. This file is located in the Theorem translator installation directory. If the location of CATIA V5 subsequently changes, the translator can be guided to the changed location by modifying this file using a text editor to modify the ***ts_env.bat*** that is located in the translator installation directory.

If no entry is included for DSLICENSING a warning dialog will be displayed which warns of the empty field. Selecting ***Yes*** to continue will allow the installation to continue.
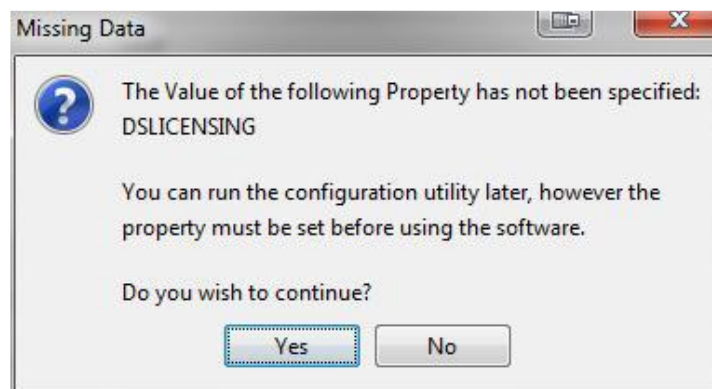
Running CATIA V5 Translators

Before running the translator the user must run CATIA V5 interactively at least once to configure the CATIA V5 environment and license settings. This can be achieved by running the catia5r<XX>_start script as follows:
<Translator_Installation_Directory>\bin\catia5r<XX>_start.cmd
Once CATIA has been run the Translator can run as described in the relevant product User Guide.

CATIA V5 Environment DIRENV & ENV

The default location for CATIA V5 to store its global environment files is in the global directory:
Windows 10:

C:\ProgramData\DassaultSystemes\CATEnv

Or

%APPDATA%\CATEnv

You can find this location by running:

%CATIAV5_INST%\<OSDS>\code\bin\setcatenv -h

The environment files are named in the form **CATIA.V5RN.B<XX>.txt**
If when installing CATIA V5 the default environment file location was replaced with another location then this location needs to be indicated to the Translator by defining in the **ts_env.bat** the environment variable CATIAV5_DIRENV:

set CATIAV5_DIRENV=/some/directory

If the Theorem installation is needed to support multiple releases of CATIA. Then the user can define release specific locations using:

set CATIAV5R<XX>_DIRENV=/some/directory

The Theorem translator will attempts to create its own environment file called **TheoremCatia5R<XX>.txt.** The user must therefore have write permission to the CATEnv directory. If this is not possible an existing environment file can be specified using the variable **CATIAV5_ENV**. e.g.

set CATIAV5_ENV=CATIA.V5R31.B31

Note. the extension **.txt** is not required. The user can specify a release specific name using **CATIAV5R<XX>_ENV** e.g.

set CATIAV5R31_ENV=CATIA.V5R31.B31

Checking the CATIA V5 Environment

A script is provided to check that the CATIA V5 environment is set up correctly. In a command window run the command script:

%TS_INST%\bin\checkcatia5r<XX>env.cmd

Checking the Theorem Shared Library

A script is provided to ensure that the CATIA V5 environment is compatible with the Theorem shared library. In a command window run the command script:

%TS_INST%\bin\checkcatia5r<XX>cadverter.cmd

A successful output is an indication that the location for CATIA V5 has been specified to the Theorem translator correctly and that the correct version of the Theorem CATIA V5 translator products have been installed.

Appendix B – 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.

If this is not supplied the following directories will be searched in the specified order for the named configuration files: (TS_INST = Installed directory)

> **tess.config** in the directory where the translator is run
> **tess.config** in TS_INST\etc directory

Two example config files are provided in the **TS_INST\etc** directory, a standard **tess.config** one, and one that illustrates the options required for large assembly processing, **tessLargeAssm.config** which is documented by some comments within it.

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	verboseReporting = false

	TRUE	
	false	
	FALSE	
writeAsciiAssembly	true	writeAsciiAssembly = false
	TRUE	
	false	
	FALSE	
singlePartsNoAssem	true	singlePartsNoAssem = false
	TRUE	
	false	
	FALSE	
smartLODgeneration	true	smartLODgeneration = true
	TRUE	
	false	
	FALSE	
autoLowLODgeneration	true	autoLowLODgeneration = true
	TRUE	
	false	
	FALSE	
numLODs	*any integer*	numLODs = 3
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 = "abc123."`
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>"
<u>AddToAllNodes</u>	*"string of characters"*	AddToAllNodes = "<metadata key>" AddToAllNodes = "<metadata value>"
<u>close brace</u>	}	}

The Level of Detail Section

The level of detail section of the configuration file contains the tessellation and simplification information for each level of detail in the file.

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

To edit level of detail options

1. Open an existing configuration file in a text editor.

2. Edit the configuration file options listed below.

3. Save the configuration with a .config extension

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

The JT Options Section

The JT Options section of the configuration file contains arguments for JT data.

To edit special options

1. Open an existing configuration file with a text editor.

2. Edit the configuration file options shown in the table below.

3. Save the configuration with a .config file extension.

Option	Keyword	Example	Default Value
Jt<u>Options</u>	`JtOptions {`	`JtOptions {`	

AddSemanticPMI	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`AddSemanticPMI = true`	true

AnnotationPlanesViewsetName	`Character string`	`AnnotationPlanesViewsetName = "Views"`	Views

AppendCADExtension	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`AppendCADExtension = false`	true

BrepType	`XT`
	`JT`
	`XTJT`
		`BrepType = true`	XT

BrepWireframe	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`BrepWireframe = false`	false

CadPropertyMappingFile	`Character String`	`CadPropertyMappingFile = "%TS_INST%\data\jt\jt_property_mapping.txt"`	

CollapseHierarchy	`SOLtoDetail`
	`toSets`
	`toPart`
	`SURandFACtoDetail`
	`toPartOptimized`
	`expandPart`
		`CollapseHierarchy = SOLtoDetail`	`SOLtoDetail`

DefaultLayerFilter	`"Character String"`	`DefaultLayerFilter = ""`	""

EnhanceColour	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`EnhanceColour = true`	true

ExpandPart	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`ExpandPart = false`	false

ExplodeSolidsToFaces	`true`	`ExplodeSolidsToFaces = false`	false

	TRUE false FALSE		
ExternalDetailNaming	Character string	ExternalDetailNaming = ""	""
FaceEdgeTol	Real	FaceEdgeTol = 0.000006	0.000006
FixDegenerateEdges	true TRUE false FALSE	FixDegenerateEdges = true	true
FixSmallFeaturesOpenSolids	true TRUE false FALSE	FixSmallFeaturesOpenSolids = false	false
FixSmallFeaturesSolids	true TRUE false FALSE	FixSmallFeaturesSolids = false	false
ForceBodyCreation	true TRUE false FALSE	ForceBodyCreation = true	true
IncrementalSewing	true TRUE false FALSE	IncrementalSewing = true	true
IncrementalSewingNoOfIterations	Integer	IncrementalSewingNoOfIterations = 5	5
JtBrepFixup	true TRUE false FALSE	JtBrepFixup = False	False
LayerFilter	true TRUE false FALSE	LayerFilter = false	false
ModelViews	true TRUE false FALSE	ModelViews = true	true
NameRootNode	true TRUE false FALSE	NameRootNode = false	false
OutputUnits	mm cm m inches	OutputUnits = mm	mm
ParasolidTolerantModelling	true TRUE	ParasolidTolerantModelling = true	true

	false FALSE		
ParasolidTolerantModellingFactor	Integer	ParasolidTolerantModel lingFactor = 3	3
PlmxmlPart	true TRUE false FALSE	PlmxmlPart = false	false
PlmxmlPropertyMappingFile	Character String	PlmxmlPropertyMappingF ile = "%TS_INST%\data\jt\plm xml_property_mapping.t xt"	
PmiAttributeMap	true TRUE false FALSE	PmiAttributeMap = false	false
PmiAttributeMapFilename	"Character String"	PmiAttributeMapFilenam e = ""	
PmiAttributeSplit	true TRUE false FALSE	PmiAttributeSplit = false	false
PmiGlyphs	2D 3D off	PmiGlyphs = 3D	3D
PmiPolygons	true TRUE false FALSE	PmiPolygons = false	false
ProduceTessellatedOutput	true TRUE false FALSE	ProduceTessellatedOutp ut = false	false
ReuseSolids	true TRUE false FALSE	ReuseSolids = false	false
SavedViewsViewsetName	"Character String"	SavedViewsViewsetName = "Catpures"	Captures
SewParasolidBodies	true TRUE false FALSE	SewParasolidBodies = true	true
SewParasolidBodiesTol	Real	SewParasolidBodiesTol = 0.01	0.01
ShineSetting	Integer	ShineSetting = 30	30
SimplifyGeometry	true TRUE false FALSE	SimplifyGeometry = false	false

SplitDiscontinuousSurfaces	true TRUE false FALSE	SplitDiscontinuousSurf aces = false	false
StructureOutputType	JT PLMXML PLMXMLJT STEPBOM STEPBOM_E2	StructureOutputType = JT	JT
SubNode	true TRUE false FALSE	SubNode = false	false
UseInstanceNames	true TRUE false FALSE	UseInstanceNames = false	false
FlushRootNodeColour	true TRUE false FALSE	FlushRootNodeColour = false	false
ExplodeSolidIntoFaces	true TRUE false FALSE	ExplodeSolidIntoFaces = false	false
AddSolidNodes	true TRUE false FALSE	AddSolidNodes = false	false
WriteSubAssyFiles	true TRUE false FALSE	WriteSubAssyFiles = false	false
create_cadid	true TRUE false FALSE	create_cadid = false	false
cadid_attr	Character String	cadid_attr = ""	
cadid_fname	Character String	cadid_fname = ""	
close brace	}	}	

The Catia5 Options Section

The Catia5 Options section of the configuration file contains arguments for Catia5 data.
To edit special options

1. Open an existing configuration file with a text editor.

2. Edit the configuration file options shown in the table below.

3. Save the configuration with a .config file extension.

Option	Keyword	Example	Default Value
Catia5Options	`Catia5Options {`	`Catia5Options {`	

ApplyTrimmingLimits	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`ApplyTrimmingLimits = false`	false

AssemblyReportFilename	`Character string`	`AssemblyReportFilename = ""`	

ConvertSurfaceFaceLimits	`On`
	`Off`
	`Powerfill`
		`ConvertSurfaceFaceLimits = On`	On

ConvertGenericFilletSurfaceTol	`Real`	`ConvertGenericFilletSurfaceTol = 0.0001`	0.0001

ConvertCurvesToNurbs	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`ConvertCurvesToNurbs = false`	false

ConvertCurvesToNurbsTol	`Real`	`ConvertCurvesToNurbsTol = 0.00001`	0.00001

ConvertSurfacesToNurbs	`None`
	`Fillets`
	`Spheres`
	`Fillets+Spheres`
		`ConvertSurfacesToNurbs = None`	None

ConvertSurfacesToNurbsTol	`Real`	`ConvertSurfacesToNurbsTol = 0.00001`	0.00001
DecimalSeparator	`dot` `comma`	`DecimalSeparator = dot`	dot

FaceOpacity	`true`
	`TRUE`
	
	`false`
	`FALSE`
		`FaceOpacity = true`	true

FeatureName	true TRUE false FALSE	`FeatureName = false`	false
GenerateEmptyPart	true TRUE false FALSE	`GenerateEmptyPart = false`	false
HybridJT	true TRUE false FALSE	`HybridJT = false`	false
InstanceNaming	`Character String`	`InstanceNaming = ""`	
MaximumNurbsDegree	`Integer`	`MaximumNurbsDegree = 5`	5
MergeCGRVertices	true TRUE false FALSE	`MergeCGRVertices = false`	false
NonManifold	true TRUE false FALSE	`NonManifold = false`	false
OpacityZero	`Real`	`OpacityZero = 0.1`	0.1
OutputMBD	true TRUE false FALSE	`OutputMBD = false`	false
OutputMBDLeafNodes	true TRUE false FALSE	`OutputMBDLeafNodes = false`	false
PrintInfo	true	`PrintInfo = true`	true

	TRUE false FALSE		
ProcessCaptures	true TRUE false FALSE	ProcessCaptures = false	false
ProcessEsplice	true TRUE false FALSE	ProcessEsplice = false	false
ProcessElectrical	true TRUE false FALSE	ProcessElectrical = false	false
ProcessHybridJT	true TRUE false FALSE	ProcessHybridJT = false	false
ProcessLargeFaces	HideLargeFaces AllowLargefaces DatumFaces True False	ProcessLargeFaces = Datumfaces	DatumFaces
ProcessLayers	ALL_LAYERS A comma separated list of layer numbers(0 - 255), using hyphen '-' to separate number ranges.	ProcessLayers = ALL_LAYERS	ALL_LAYERS
ProcessMotion	true	ProcessMotion = false	false

	TRUE false FALSE		
ProcessOpenSolids	true TRUE false FALSE	`ProcessOpenSolids = true`	true
ProcessPMI	true TRUE false FALSE	`ProcessPMI = false`	false
ProcessPoints	true TRUE false FALSE	`ProcessPoints = true`	true
ProcessSolids	true TRUE false FALSE	`ProcessSolids = true`	true
ProcessWelds	true TRUE false FALSE	`ProcessWelds = false`	false
ProcessWireFrame	true TRUE false FALSE	`ProcessWireFrame = true`	true
ReadCachedCGR	true TRUE false FALSE	`ReadCachedCGR = false`	false
ReadCGREdges	true	`ReadCGREdges = true`	true

	TRUE false FALSE		
ReadFaceColours	true TRUE false FALSE	ReadFaceColours = true	true
ReadFTAReferenceGeometry	true TRUE false FALSE	ReadFTAReferenceGeometry = false	false
ReadHiddenData	None Geometry Structure All	ReadHiddenData = None	None
ReadInfiniteAxis	true TRUE false FALSE	ReadInfiniteAxis = false	false
ReadLeafNodeCGR	true TRUE false FALSE	ReadLeafNodeCGR = true	true
ReadMotion	true TRUE false FALSE	ReadMotion = false	false
ReadNOA	true TRUE false FALSE	ReadNOA = false	false
ReadScanData	true TRUE	ReadScanData = true	true

	false FALSE		
ReadSpaceReservations	true TRUE false FALSE	ReadSpaceReservations = false	false
ReduceNurbsSurfaceDegree	true TRUE false FALSE	ReduceNurbsSurfaceDegree = false	false
RetainAssemblyStructure	true TRUE false FALSE	RetainAssemblyStructure = true	true
RetainTrailingZeros	true TRUE false FALSE	RetainTrailingZeros = false	false
SimplifyCurves	true TRUE false FALSE	SimplifyCurves = false	false
SimplifyProgressFile	true TRUE false FALSE	SimplifyProgressFile = false	false
SelectionSets	true TRUE false FALSE	SelectionSets = false	false
SurfaceChecking	true TRUE false	SurfaceChecking = false	false

Option	Keyword	Example	Default Value
	FALSE		
SurfaceCheckingTol	Real	SurfaceCheckingTol = 0.001	0.001
TextName	true TRUE false FALSE	TextName = false	false
TopGroupPerBody	true TRUE false FALSE	TopGroupPerBody = false	false
TrimFaceSurfaces	true TRUE false FALSE	TrimFaceSurfaces = true	true
UDFAxisSystems	true TRUE false FALSE	UDFAxisSystems = false	false
ViewContext	Default3D Unfolded	ViewContext = Default3D	Default3D
close brace	}	}	

The Catia4 Options Section

The Catia4 Options section of the configuration file contains arguments for Catia4 data.

To edit special options

1. Open an existing configuration file with a text editor.

2. Edit the configuration file options shown in the table below.

3. Save the configuration with a .config file extension.

Option	Keyword	Example	Default Value
Catia4Options	Catia4Options {	Catia4Options {	

CatiaAssemblyPartsPath	ASIS SET LAYER ENTITYTYPE VIEW	ColourVisualizationMode = ASIS	ASIS
ColourVisualizationMode	Character string	AssemblyReportFilename = ""	
FacetColour	true TRUE false FALSE	FacetColour = false	false
GroupPipes	groupPipelines groupPipes ungroupPipelines	GroupPipes = groupPipelines	groupPipelines
ProcessEntityNames	ALL_ENTITIES A comma separated list of ENTITY names	ProcessEntityNames = ALL_ENTITIES	ALL_ENTITIES
ProcessFAC	true TRUE false FALSE	ProcessFAC = true	true
ProcessGPR	true TRUE false FALSE	ProcessGPR = true	true
ProcessLayers	ALL_LAYERS CURRENT_LAYER_FILTER A comma separated list of layer numbers(0 - 255), using hyphen '-' to separate	ProcessLayers = ALL_LAYERS	ALL_LAYERS

	number ranges.		
ProcessModelNumber	Integer	ProcessModelNumber = 1	1
ProcessNOSHOW	true TRUE false FALSE	ProcessNOSHOW = false	false
ProcessNRBS	true TRUE false FALSE	ProcessNRBS = true	true
ProcessPIP	true TRUE false FALSE	ProcessPIP = true	true
ProcessPOI	true TRUE false FALSE	ProcessPOI = true	true
ProcessSetNames	ALL_SETS A comma separated lists of SET names	ProcessSetNames = ALL_SETS	ALL_SET S
ProcessSKI	true TRUE false FALSE	ProcessSKI = true	true
ProcessSOL	true TRUE false FALSE	ProcessSOL = true	true
ProcessSTR	true TRUE	ProcessSTR = true	true

	false		
	FALSE		
ProcessSUR	true	ProcessSUR = true	true
	TRUE		
	false		
	FALSE		
ProcessWireFrame	true	ProcessWireFrame = true	true
	TRUE		
	false		
	FALSE		
UseColorMethod	AS_FILED	UseColorMethod = AS_FILED	AS_FILED
	SET		
	LAYER		
	ENTITY		
close brace	}	}	

The General Options Section

The General Options section of the configuration file contains general arguments.

To edit special options

1. Open an existing configuration file with a text editor.

2. Edit the configuration file options shown in the table below.

3. Save the configuration with a .config file extension.

Option	Keyword	Example	Default Value
GeneralOptions	GeneralOptions {	GeneralOptions {	
MassProperties	true	MassProperties = false	false
	TRUE		
	false		
	FALSE		
ProgressFileInformation	true	ProgressFileInformation = true	true
	TRUE		
	false		

	FALSE		
ProgressFileWarnings	true TRUE false FALSE	ProgressFileWarnings = true	true
ReportFilename	Character String	ReportFilename = "%TEMP%\report1.log"	
AdvancedOptions	Chatacter String	AdvancedOptions = ""	
close brace	}	}	

Appendix C – Error Codes

Common Error Codes

When processing data, the command screen or log file will show a return code as a status of the translation. These are detailed below.

Return Code	Description
0	Success
1	Issues during write leg such as incomplete run or no arguments given
2	Issue obtaining license
3	Input file argument issue or failure to open Progress File
4	Issue during read including no entities read
5	Failure to located secondary process (aka vwr_3dpdf.exe or vwr_jt.cmd)
6	
7	Failure to open CATIA V5 document
8	Failure to find root node of CATIA V5 document
9	Failure to start CATIA V5 session
10	catia5cadverter.dll version doesn't match the catia5_XXX.exe
11	CATIA V4 data given as input file but bi _V4MAIN license available

THEOREM
SOLUTIONS

UK, Europe and Asia Pacific Regions

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

sales@theorem.com

+44 (0) 1827 305 350

USA and the America

THEOREM SOLUTIONS INC
100 WEST BIG BEAVER
TROY
MICHIGAN
48084
USA

Sales-usa@theorem.com

+(513) 576 1100

THEOREM.COM