TRANSLATE CATIA V5i <> JT

USER GUIDE

Revision: 8.0

Issued 04/09/2023

Document History

Version	Description	Date Updated
1.0	Initial Release	11/08/2016
2.0	Updated for 20.3 release	18/01/2018
3.0	Updated for 21.2 release	02/08/2018
4.0	Updates for 22.2 release	28/08/2019
5.0	Updates for 23.1 release	30/04/2020
6.0	Updated for 24.1 release	03/03/2022
7.0	Updated for 25.1 release	26/05/2022
8.0	Updated for 26.0 release	04/09/2023

Contents

Overview of TRANSLATE

About Theorem

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

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

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

Theorem's Product Suite

Theorem have 3 main Product brands. These are:

TRANSLATE

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

See our website for more detail.

PUBLISH

The creation of documents enriched with 3D content

See our website for more detail.

VISUALIZE

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

See our website for more detail.

The CATIA V5i Bi-directional JT Translator

The CATIA V5i to JT Translator is a direct database converter between CATIA V5 and JT. It enables the user to convert all forms of mechanical design geometry, as well as assembly and attribute information, between these two systems without requiring access to a CATIA V5 license.

The Translator can be purchased as a uni-directional, CATIA V5 to JT, or JT to CATIA V5 product, or as a bi-directional product.

It is possible to invoke the translator in batch mode with the command line interface allowing the conversion process to be integrated into any process-oriented operation. Alternatively the conversion process may be operated by using the Theorem Unified Interface.

Primary Product Features

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

- If assembly data (product structure) is in the file, the assembly structure will be mapped between the two systems as well as colour information

- Converts attribute data including colour and layer information.

- The user can filter data to optimize the process.

- Uses the Spatial CGM 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.

- If you wish to visualise and interrogate the CATIA V5 or JT data this can be done by using the integrated User Interface, which is included with the translator.

- There is no dependency on a CATIA V5 installation or application license.

- The conversion process can be in Batch Mode or using the Unified Interface.

- Command line interface allows process integration.

Primary Product benefits?

- Direct conversion between CATIA V5 and JT reduces processing time, simplifies integration and retains accuracy of the model

- The integrated viewing capability enables visual verification, pre and post translation

- The integrated data filtering options allows selected data ONLY to be processed, enabling optimisation of translations and time savings

- By converting all forms of geometry no data is lost, eliminating the time required to recreate missing data

- With over 20 years industrial use Theorem's product robustness and quality is well proven, reducing your business risk

- This document will focus specifically on guidance for the use of the translator for CATIA V5i to 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
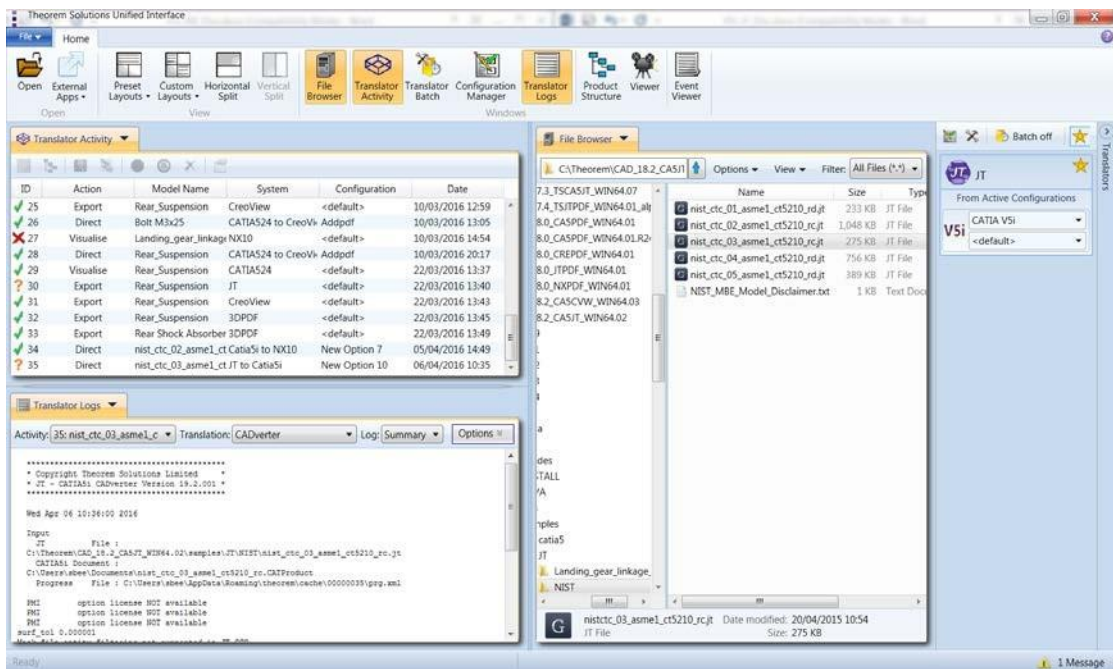
Using the Product

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.
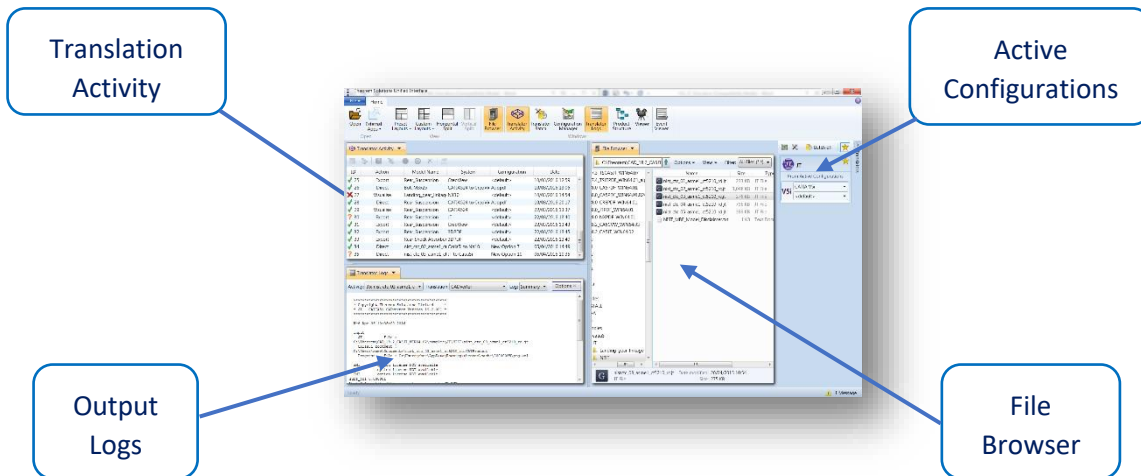
Alternatively, the Unified Interface can be run via a Windows Explorer selection in:

<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:

Translation Activity

Active Configurations

Output Logs

File Browser

©Theorem Solutions 2022

The simplest way to translate from CATIA V5i to JT is to drag a file from the file Browser Pane on to the Active Configurations for the translation you require.
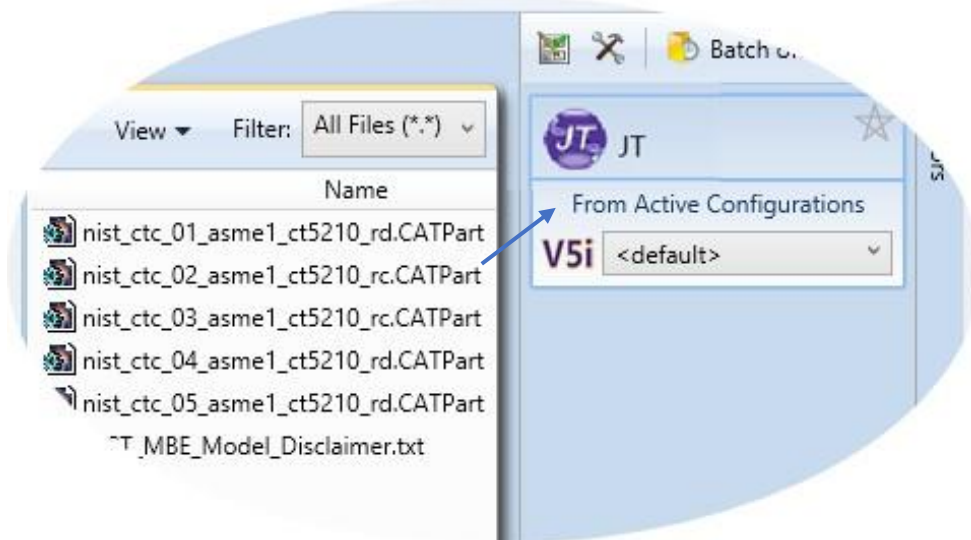
On completion, the Unified Interface will display the activity information and details from the log file created during the translation, if requested, in the Translation Activity and Output Log panes, respectively.

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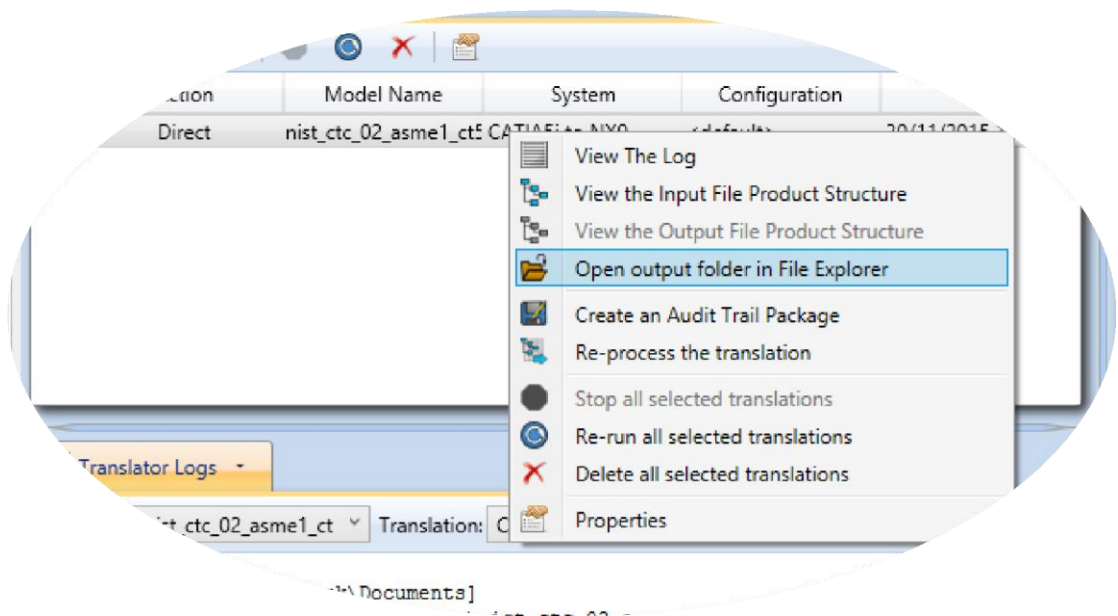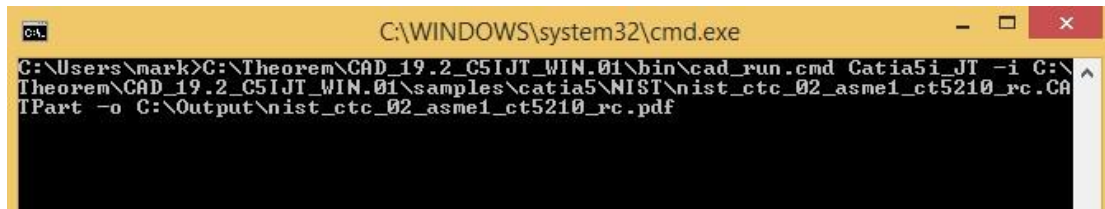
Default Translation – via the Command Line

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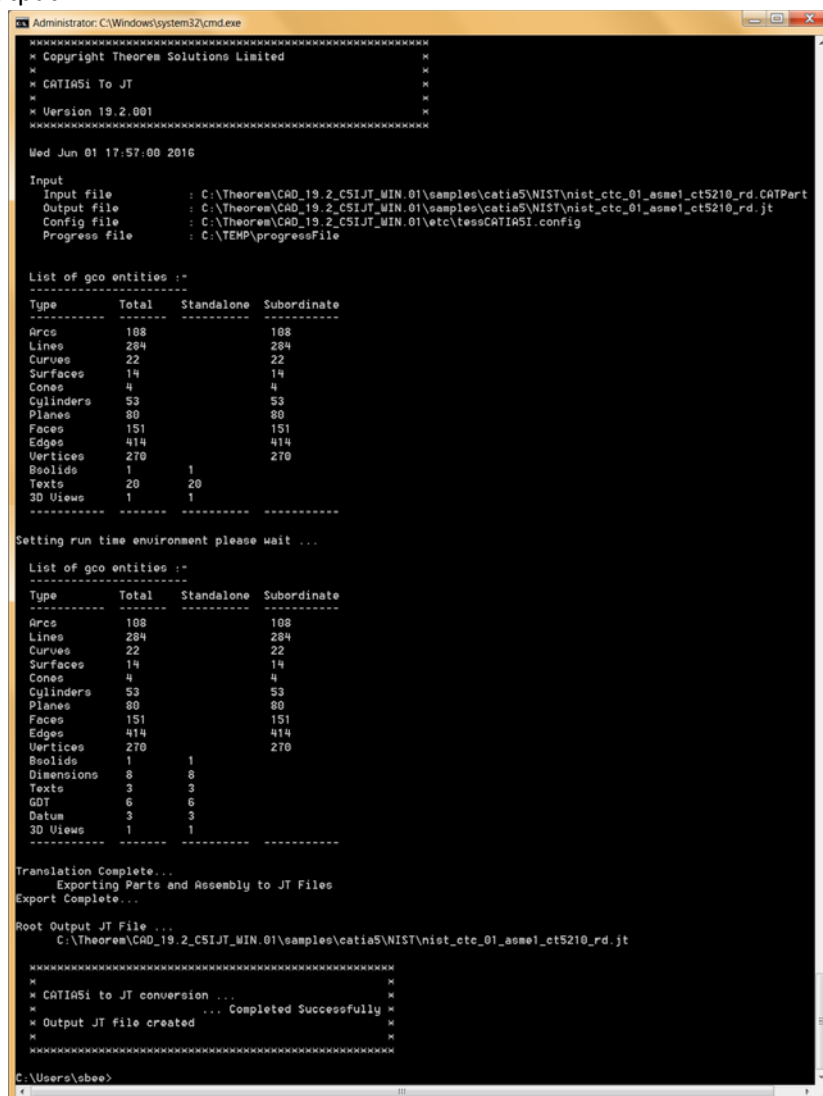
The format of the command is as follows when translating from CATIA V5 to JT:

<Translator_installation_directory>\bin\cad_run.cmd Catia5i_jt –i <input_file> -o <output_file>

The example above will translate a CATIA V5 sample file provided within the installation and produce the following screen output:

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

%Theorem_Installation%\samples\catia5\NIST\nist_ctc_03_asme1_ct5210_rc.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 V5i 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 the following areas:

- CATIA V5i 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

- Masking   - Additional Read/Write options to limit the types of data translated e.g. Solids Only

- General   - Those arguments that are common to ALL Publishing activities regardless of source
  data

CATIA V5i Read Arguments

The image below shows the CATIA V5I Read arguments that are available, with their default settings:

Each of these options is described below:

Option	Description

Retain Assembly Structure	Retain the assembly structure. Default is ON.
	
	  o Command Line Syntax
	
	    ▪ *offditto* (to disable) – reduces an assembly to a single Part

Read Face Colours	Process face colours in preference to body colours. Default is OFF
	
	  o Command Line Syntax
	
	    ▪ *disable_face_colours*

Read Hidden Data	Allow selective data types to be read regardless of hide/show state. Default is OFF.
	
	  o Command Line Syntax
	
	    ▪ *read_hidden_geometry*

Read PMI	Reads any PMI data available in the V5 file(s)

Categorise PMI	PMI is categorised by type. Default is ON. Only selectable if *'Read PMI'* is selected. This can be switched off using
	
	  o Command Line Syntax
	
	    ▪ *disable_catagorise_pmi*

Read Hidden Views	Reads any Views that are hidden

JT Write Arguments

The image below shows the JT Write arguments that are available, with their default settings:
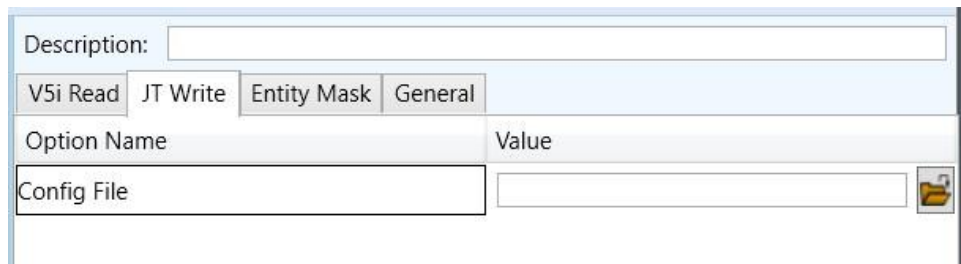
Each of these options is described below:

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

CATIA V5i to JT Entity Mask Arguments

The image below shows the CATIA V5i to JT Entity Mask arguments that are available, with their default settings:

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:
	    "SOL" - Masks any 3D entity
	    "SKIN" - Masks any 2D entity
	    "CUR" - Masks any 1D entity
	    "POI" - Masks any 0D entity
	
	    "AXIS" - Masks Axis Systems
	    "ISOL" - Masks Isolated facetted solids
	    "CCRV" - If on creates a CCRV curve for wire frame edges that
	           have more than one supporting curve
	    "TEXT" - Masks PMI Text
	
	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 = Off)* o Command Line Syntax: ■ *Add the following entry to the Mask file*
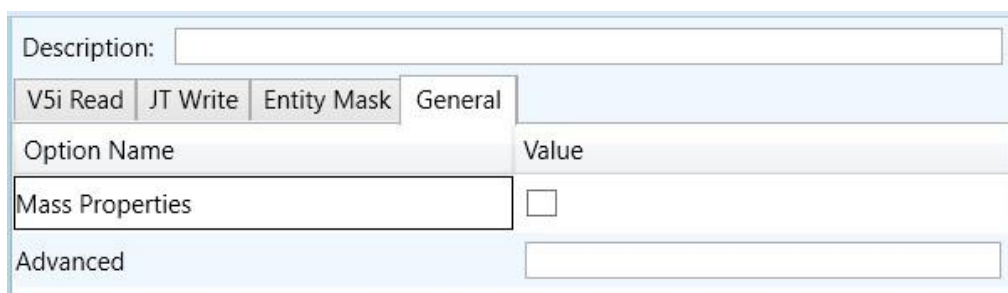
	ON NOSHOW GEO
Convert No Show Structure	Enables Hidden Assembly Structure to be processed *(Default = Off)* o Command Line Syntax: ▪ *Add the following entry to the Mask file* **ON NOSHOW STR**
Convert No Show PMI	Enables Hidden PMI to be processed *(Default = Off)* o Command Line Syntax: ▪ *Add the following entry to the Mask file* **ON NOSHOW PMI**

CATIA V5i to JT General Arguments

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	CATIA V5 mass properties (volume/area CofG) are read and any applied materials, using this option, in cases where a part has multiple solids, volume and area values are summed, but CofG data is invalid. o Command Line Syntax ▪ *mprops*
Advanced	Allows any of the Command Line Advanced arguments documented to be passed to the Unified Interface invocation.

CATIA V5i to JT Advanced Arguments

Theorem's CATIA V5i to JT translator has been configured with default settings that optimises the translation process. However, there are times when a satisfactory result cannot be obtained, so it may be required to deploy one or more Advanced Arguments to improve the translated result.

Option	Description
Simplify Curves	Convert NURBS curves to conics. Default is OFF. ○ Command Line Syntax ▪ *simplify_curve*
Convert Curves to NURBS	Convert curves to NURBS. Default is OFF. ○ Command Line Syntax ▪ *convert_curves*
Conversion Tolerance	A secondary argument to *'Convert Curves'* defining the conversion tolerance. Default is 0.00001 ○ Command Line Syntax ▪ *convert_curve_tol 0.00001*

Convert Surfaces to NURBS	Process data (read) types as NURBS. Data type is selected from options. Default is Fillets.
	○ Command Line Syntax
	   ▪ *None: dont_convert_fillets*
	   ▪ *Fillets: Default Option.*
	   ▪ *Spheres: dont_convert_fillets convert_spheres*
	   ▪ *Fillets + Spheres: convert_spheres*
	   ▪ *All: convert_surfaces*

Convert Torus to NURBS	Even when data is read as NURBS data, the Torus types are converted to NURBS by default, this can be disabled using the command line ○ Command Line Syntax ▪ *dont_convert_torus*
Conversion Tolerance	A secondary option to 'Convert Surfaces to NURBS'. Defines the conversion tolerance. Default is 0.00001. ○ Command Line Syntax ▪ *convert_surface_tol 0.00001*
Trim Face Surfaces	*Trims face surfaces. Default is ON.* ○ *Command Line Syntax* ▪ *dont_trim_surfaces*
Process Large Faces	Enable reading of faces larger than 1km. Default is OFF. ○ Command Line Syntax ▪ allow_large_faces
UDF Axis Systems	Enable reading of User Defined Axis systems. Default is OFF. ○ Command Line Syntax ▪ *read_udf_axis – to turn on*

Graphical Read	By default the BREP data will be read. It is possible to read the CATIA V5 data as a graphical representation using this option
	    o Command Line Syntax
	        ▪ *enable_graphical*

Filter Geometry	It is possible to filter large planes (construction planes) larger than a given size using (default being 1000 meters)
	    o Command Line Syntax
	        ▪ *filter_large_geom <meters>*
	
	There is a special case for PLANES (typically construction planes) which are by defult not read, these can be enabled using
	    o Command Line Syntax
	        ▪ read_planes

Disable the read of Geometry	By default all entities are read, the following options allow the masking of specified data types:
	    o Command Line Syntax:
	        ▪ *disable_solids*
	        ▪ *disable_skins*
	        ▪ *disable_wireframe*
	        ▪ *disable_points*
	        ▪ *struct_read - only read assembly structure, everything else being disabled*

Disable the read of meta data	By default all attributes are read, the following options allow the masking of this data:
	    o Command Line Syntax:
	        ▪ *disable_attributes*

Parasolid Tolerant Modelling	A secondary option to be used when Brep Type = XT Brep (Theorem)
	output is specified. Enables Parasolid tolerant modelling. Default is ON
	    o Command Line Syntax
	        ▪ nopstolmodel – to turn off

Factor	Specify the factor level of Parasolid Tolerant Modelling when turned on. Default is 3.
	    o Command Line Syntax
	        ▪ pstolmodel 3

Sew Parasolid Bodies	A secondary option to be used when Brep Type = XT Brep (Theorem)
	output is specified. Enables the sewing of Parasolid bodies. Default is ON
	    o Command Line Syntax
	        ▪ nosew – to turn off

Tolerance	Specify the tolerance for the sew command above. Default is 0.01.
	    o Command Line Syntax
	        ▪ pssew 0.01

Incremental Sewing	Enables incremental sewing when used with Sew Parasolid Bodies.
	Default is ON.
	     o Command Line Syntax
	           ▪ no_sew_increm – to turn off

Split Discontinuous Surfaces	A secondary option to be used when Brep Type = XT Brep (Theorem)
	 output specified. Splits discontinuous surfaces.
	Default is OFF.
	     o Command Line Syntax
	           ▪ brep_prep – to turn on
	           ▪ no_brep_prep – to turn off

**Force body creation
(No check of Parasolid entities)**
	A secondary option enabled when XT Brep (Theorem)
	output specified. Removes the checking of Parasolid entities.
	Default is ON.
	     o Command Line Syntax
	           ▪ *nocheck* – (force body creation without checking = Default)
	           ▪ *check* – (doesn't force the body creation - Parasolid checking is enabled)

Fix Degenerate Edges	A secondary option to be used when *Brep Type = XT Brep (Theorem)*
	output specified. On face create failure, check and fix any degenerate edges.
	 Default is ON.
	     o Command Line Syntax
	           ▪ *fix_degen*
	           ▪ *no_fix_degen – to turn off*

Specify a Face Edge Tolerance	A secondary option to be used when *Brep Type = XT Brep (Theorem)*
	output specified. Specify an edge tolerance to be used when creating faces.
	Default is ON.
	     o Command Line Syntax
	           ▪ *Please see Edge Tolerance below*

Edge Tolerance	A secondary option used with Specify a Face Edge Tolerance where the
	tolerance value is assigned.
	Default is 0.000006.
	     o Command Line Syntax
	           ▪ *face_edge_tol 0.000006*

Fix small features in solids	A secondary option to be used when *Brep Type = XT Brep (Theorem)* output specified. Remove small edges, sliver and spike faces from solid bodies. Default is OFF.

	o Command Line Syntax
	    ▪ *ps_fix_small – to turn on*
	    ▪ no_ps_fix_small - default

Fix small features in open solids	A secondary option to be used when *Brep Type = XT Brep (Theorem)*
	output specified. Remove small edges, sliver and spike faces from open solids.
	Default is OFF.
	    o Command Line Syntax
	        ▪ *ps_fix_osol – to turn on*
	        ▪ *no_ps_fix_osol - default*

Simplify Geometry	A secondary option to be used when *Brep Type = XT Brep (Theorem)*
	output specified. Simplify Geometry.
	Default is OFF.
	    o Command Line Syntax
	        ▪ *simplify_solids – to turn on*

Expand Part	This option will generate a tree structure for each solid in the part
	    o Command Line Syntax
	        ▪ expand_part

Common Options for JT to CATIA V5i

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

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

- Catia5i Write – Those arguments that affect how the data is written to Catia5

- General – Those arguments that are common to ALL Publishing activities regardless of source data

JT Read Arguments

The image below shows the CATIA V5I Read arguments that are available, with their default settings:

Each of these options is described below:

Option	Description

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*

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 V5i Write Arguments

The image below shows the CATIA V5I Write arguments that are available, with their default settings:

Each of these options is described below:
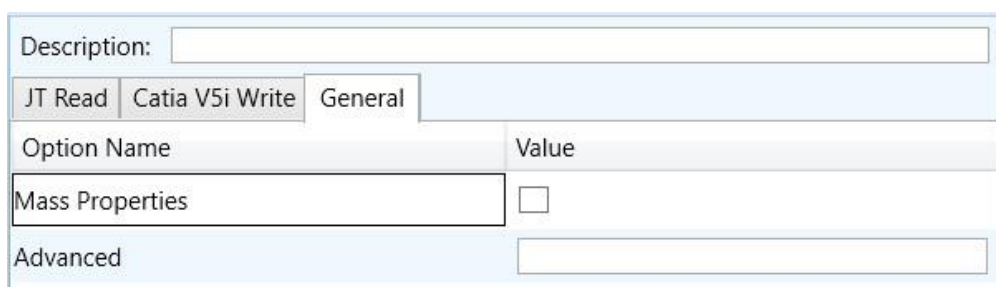
Option	Description

Save Catia5 Version	Save a specified version of Catia5 data (default 25 (2015))
	
	    o Command Line Syntax
	       ▪ *save_catia5_version <version>*
	
	 Where versions are :
	
	       ▪ 16-32
	       ▪ 2012-2022

Retain Assembly Structure	The default processing will write any JT assembly structure to CATProduct structure files with each JT part being written as separate CATIA V5 CATPart or CGR output. When this option is deselected the output representation will be a single file (CATPart or CGR) dependent upon the format selection The command line option off_ditto would achieve the same results.
Create CGR	Selecting this option will write the geometry as CGR format data, rather than the default CATPart representation of the output geometry. The version of the CGR format will be consistent with the highest version of CATIA V5 output supported with the current version of V5 libraries. To create CGR data suitable for earlier releases of CATIA V5 select the version of the output required from the menu selector.

JT to CATIA V5i General Arguments

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	CATIA V5 mass properties (volume/area CofG) are read and any applied materials, using this option, in cases where a part has multiple solids, volume and area values are summed, but CofG data is invalid.
	
	    o Command Line Syntax
	         ▪ *mprops*

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

The available Advanced options are described below:

JT to Cata5i Advanced Arguments

Advanced Option	Description
Solid Colour	When writing CATIA data from JT the colour of the CATIA V5 output can be specified to be taken from the JT colour associated with the solid by using the option **solid_colour**. The default would be to use the colour associated with the JT faces rather than the colour applied at the solid level.
Occurrence Tree Output	When processing JT assemblies that have had unique colours applied at the occurrence of multiply instanced parts it is necessary to expand the CATIA V5 assembly output to enable the occurrence colour to be applied. Using the option **occur_tree** will expand the assembly structure and generate a unique occurrence for each instance of the assembly structure.

Small Curves	Report Small curves as errors. (default OFF)
	
	    o Command Line Syntax
	        ▪ small_curves (to enable)

Extend Nurb Surfaces	Extends NURBS surfaces beyond face limits for curve projection (default state)
	
	    o Command Line Syntax
	
	        ▪ no_extend_nurb -(*Don't extend NURBS surfaces to face limits*)
	        ▪ extend_nurb <int> - (trims NURBS surfaces to <int> * 0.0001 face extents in u and v)

Remove Groups	Remove Group entities into assembly structure. (default OFF)
	
	    o Command Line Syntax
	        ▪ remove_groups (to enable)

Use Ref Name	Uses file name from input system to name files (default OFF)
	
	    o Command Line Syntax
	        ▪ use_ref_name (to enable)

Re-tessellate Brep Data	Allows Brep data to be re-tessellated to create a link between the faces in the Brep and the tessellated

	representation. There is a choice between processing just the tessellated form or both the tessellated and Brep forms with the links between them. Default is 'Off'.
	
	 o Command Line Syntax
	
	   ▪ *On – Re-tessellate Brep Data: tess_brep*
	   ▪ *On – Re-tessellate and Read Brep Data: tess_and_read_brep*

Read as Bounding Boxes	This option allows the user to read the JT file as a very light weight bounding box representation, with each solid having its own bounding box. This option may be useful when the part's details are not of interest, as much as the <u>physical space</u>, the part occupies. o Command Line Syntax ▪ *bounding_box*
Exclude small solids (by absolute value)	Exclude solids that have bounding boxes smaller than value. Default is OFF. o Command Line Syntax ▪ *bb_exclude_value*
Exclude small solids (by bounding box fraction)	This option allows the user to exclude small solids from the JT read, the excluded solids being smaller than a specified fractional size of the overall bounding box of the part. o Command Line Syntax ▪ *bounding_box_frac [value]*
Exclude parts File for bounding box checks	This option allows the user to supply an input file containing a list of JT parts, one per line. The parts specified in the file will be excluded from the checks to remove small solids. This option is used in conjunction with the "Exclude Small Solids" options. o Command Line Syntax ▪ *bb_exclusion_file '[path_to_file]*

Appendix A – JT Configuration File

Introduction

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

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

tessCATIA5i.config in the directory where the translator is run **tess.config** in

the directory where the translator is run **tessCATIA5i.config** in TS_INST\etc

directory **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	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 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	Key Words	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>"`

AddToAllNodes	`"string of characters"`	`AddToAllNodes = "<metadata key>"`
		
		`AddToAllNodes = "<metadata value>"`

close brace	`}`	`}`

The Special Section

The special section of the configuration file contains lines that are unique to this translator.

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
Catia5iOptions	`Catia5iOptions {`	`Catia5iOptions {`	
ReportFilename	`Full system file path`	`Windows example ReportFilename = P:\caddata\translation\result\ part55`	Windows system C:%TEMP%\ tscprogressyi

OutputUnits	`mm millimetres cm centimetres m metre metres inches feet yards`
	
	`inputUnits`
		`OutputUnits = mm`	inputUnits

process_hidden_geom	`true/TRUE false/FALSE`	`process_hidden_geom = false`	false
process_hidden_views	`true/TRUE false/FALSE`	`process_hidden_views = false`	false
disable_wireframe	`true/TRUE false/FALSE`	`disable_wireframe = false`	false
disable_point	`true/TRUE false/FALSE`	`disable_point = false`	false
disable_attributes	`true/TRUE false/FALSE`	`disable_attributes = false`	false
disable_surfaces	`true/TRUE false/FALSE`	`disable_surfaces = false`	false
disable_solids	`true/TRUE false/FALSE`	`disable_solids = false`	false
disable_face_colours	`true/TRUE false/FALSE`	disable_face_colours = true	false

disable_pmi	`true/TRUE` `false/FALSE`	`disable_pmi = true`	false
disable_categorise_ pmi	`true/TRUE` `false/FALSE`	`disable_ categorise_pmi = true`	false
read_udf_axis	`true/TRUE` `false/FALSE`	read_udf_axis = true	false
read_planes	`true/TRUE` `false/FALSE`	read_planes = true	false
struct_read	`true/TRUE` `false/FALSE`	`struct_read = false`	false
StructureOutputType	JT PLMXML PLMXMLJT	StructureOutputType = JT	JT
PLMXMLProperty MappingFile	*File Name*	PLMXMLPropertyMappingFile = "PLMXML_mapping_file.txt"	"%TS_INST%\ data\jt\plmx ml_property _mapping.txt"
CadPropertyMappi ngFile	*File name*	CadPropertyMappingFile = "%CAD_mapping_file.txt"	"%TS_INST%\ data\jt\v5_jt _property_m apping.txt"
brepType	XT JT XTJT	brepType = XT	JT
ParasolidTolerant Modelling	true/TRUE false/FALSE	ParasolidTolerantModelling = true	true
ParasolidTolerant ModellingFactor	*Any positive integer*	ParasolidTolerantModelling Factor = 3	3
SewParasolidBodie s	true/TRUE false/FALSE	SewParasolidBodies = true	true
SewParasolidBodie sTol	*Any number*	SewParasolidBodiesTol = 0.01	0.01
IncrementalSewing	true/TRUE false/FALSE	IncrementalSewing = true	true
IncrementalSewing NoOfI terations	true/TRUE false/FALSE	IncrementalSewingNoOfIteratio ns = 5	5
ExplodeSolidstoFac es	true/TRUE false/FALSE	ExplodeSolidstoFaces = false	false
SplitDiscontinuous Surfaces	true/TRUE false/FALSE	SplitDiscontinuousSurfaces = true	false

ForceBodyCreation	true/TRUE false/FALSE	ForceBodyCreation = true	true
FixDegenerateEdges	true/TRUE false/FALSE	FixDegenerateEdges = true	true
FaceEdgeTol	*Any number*	FaceEdgeTol = 0.000006	0.000006
FixSmallFeaturesSolids	true/TRUE false/FALSE	FixSmallFeaturesSolids = false	false
FixSmallFeaturesOpenSolids	true/TRUE false/FALSE	FixSmallFeaturesOpenSolids = false	false
SimplifyGeometry	true/TRUE false/FALSE	SimplifyGeometry = false	false
BrepWireframe	true/TRUE false/FALSE	BrepWireframe = true	true
ProduceTessellatedOutput	true/TRUE false/FALSE	ProduceTessellatedOutput = false	false
ExpandPart	true/TRUE false/FALSE	ExpandPart = false	false
ReuseSolids	true/TRUE false/FALSE	ReuseSolids = false	false
CADPropertyMappingFile	*File Name*	CADPropertyMappingFile = "mapping_file.txt"	""
SavedViewsViewSetName	*"string of characters"*	SavedViewsViewSetName	"SavedViews"
AnnotationPlanesViewSetName	*"string of characters"*	AnnotationPlanesViewSetName	"AnnotationPlanes"
close brace	}	}	

Appendix B – Property Mapping Files

Property mapping files are required for CAD property Mapping and PLMXML Property

Mapping

A Property Mapping File is a comma separated text file containing information of how CAD properties from the source system will be mapped into the target file.

<u>The format is as follows:</u>

- • <u>Lines</u> beginning with a "#" are treated as comment lines and are ignored.
- • Any space characters will be treated as part of the item
- • Lines containing a mapping must contain 6 items separated by 5 commas

    The six items are :-

Item	Description
Source name	The attribute name in the Source System
Target name	The attribute name in the Target File
Data derived from	0 - Do not convert 1 - Use the source value as given 6 - Use the source value as given and hide the property *Note! Value 6 For CAD Mapping Files ONLY (Not PLMXML)*
Default Value	Not currently used
Value Type	Not currently used
Default Units	Not currently used

An Example of a mapping file is shown below:-

Mapping from input attribute name to Target property name
#
Line Format:-
Source name,Target name,Data derived from,Default Value,Value Type,Default Units #   Data derived from:-
0 - Do not convert
1 - Use the source values as given
6 - Use the source value as given and hide the property
#
_ActivateBOM,NULL,0,0,,
_LastModifier,NULL,0,0,,
_Maturity,NULL,0,0,,
_PrdVersion,NULL,0,0,,
_ReferenceTimeStamp,NULL,0,0,,
_Responsible,NULL,0,0,,
COG M,ud_CAD_CENTER_OF_GRAVITY,1,0,, COMPONENTS PRINCIPAL
AXES ,NULL,0,0,,
DENSITY Kg/M^3,NULL,0,0,,
INERTIA MATRIX KgM2,ud_CAD_MOMENT_OF_INERTIA,1,0,,
INERTIA VOLUME M^3,ud_CAD_VOLUME,1,0,,
INERTIA WET AREA M^2,ud_CAD_SURFACE_AREA,1,0,,
MASS Kg,ud_CAD_MASS,1,0,,
PRINCIPAL MOMENTS KgM^2,NULL,0,0,,
FILESAVETIME,File Last Modified,1,0,,
LOCALE,LOCALE,1,0,,
Masterdata Version,Masterdata Version,1,0,,
Material Details,Material Details,1,0,,
PART_NUMBER,PART_NUMBER,1,0,,
MPARTNAME,Source Model Name,1,0,,
Source,SourceName,1,0,,