



User Guide Catia V5 <> Creo

Product Category	CADTranslate
Product Group	Catia V5 <> Creo
Product Release Version	27.1

Document Type	User Guide
Document Status	Released
Document Revision	1.0
Document Author	Bruce Pittman
Document Issued	11/02/2025

📍 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

Overview of TRANSLATE.....	4
<i>About Theorem</i>	<i>4</i>
<i>Theorem’s Product Suite</i>	<i>5</i>
CADTranslate	5
CADPublish.....	5
TheoremXR	5
<i>Primary Product Features.....</i>	<i>6</i>
<i>Primary Product benefits?</i>	<i>6</i>
Getting Started	7
<i>Documentation.....</i>	<i>7</i>
<i>Installation</i>	<i>7</i>
<i>License Configuration</i>	<i>7</i>
<i>Using the Product.....</i>	<i>7</i>
Using the Product.....	8
<i>Default Translation – via the Unified Interface</i>	<i>8</i>
<i>Default Translation – via the Command Line.....</i>	<i>10</i>
<i>Default Translation – via the Interactive method</i>	<i>11</i>
CADverter Customization	11
<i>Common Options for CATIA V5 to CREO.....</i>	<i>11</i>
CATIA V5 Read Arguments.....	11
CREO Write Arguments	13
CATIA V5 to CREO Entity Masking Arguments.....	14
CATIA V5 to CREO General Arguments.....	15
<i>Common Options for CREO to CATIA V5.....</i>	<i>16</i>
CREO Read Arguments	16
CATIA V5 Write Arguments.....	18
CREO to CATIA V5 General Arguments.....	19
<i>Command Line Advanced Arguments</i>	<i>21</i>
Creo Arguments	21
CATIA V5 Arguments	22
Large Assembly Processing	23
<i>Overview.....</i>	<i>23</i>
LAP Advantages:.....	23
LAP Disadvantages:	23
<i>LAP Process Overview</i>	<i>23</i>
<i>CATIA V5 to Creo.....</i>	<i>24</i>
Command Line Arguments.....	24
Example Wrapper Script:.....	24
Updating the Assembly	24
Troubleshooting Files.....	25
<i>Creo to CATIA V5.....</i>	<i>26</i>

Command Line Arguments.....	26
Example Wrapper Script:.....	26
Updating the Assembly	26
Appendix A – CATIA V5 Configuration	28
<i>Introduction.....</i>	<i>28</i>
<i>Conventions.....</i>	<i>28</i>
Release of CATIA V5.....	28
Platform specific directory.....	28
Theorem Installation directory	28
<i>CATIA V5 Installation Directory.....</i>	<i>28</i>
<i>Running CATIA V5 Translators.....</i>	<i>29</i>
CATIA V5 Environment DIRENV & ENV	29
<i>Windows XP:</i>	<i>29</i>
<i>Windows 7, 8 & 10:</i>	<i>29</i>
Checking the CATIA V5 Environment.....	30
Checking the Theorem Shared Library.....	30

Overview of TRANSLATE

About Theorem



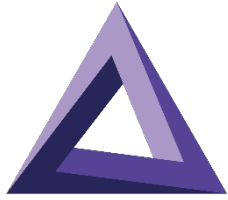
Theorem Solutions is a world leader in the field of Engineering Data Services and Solutions. This leadership position stems from the quality of our technology and the people in the company. Quality comes not only from the skills and commitment of our staff, but also from the vigorous industrial use of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



CADTranslate

CADTranslate

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

See our [website](#) for more detail.



CADPublish

CADPublish

The creation of documents enriched with 3D content

See our [website](#) for more detail.



TheoremXR

TheoremXR

Visualization for [Augmented \(AR\)](#), [Mixed \(MR\)](#) and [Virtual \(VR\)](#) Reality applications

See our [website](#) for more detail.

Primary Product Features

- CADverter converts all geometry
- If assembly data (product structure) is in the file, the assembly structure will be mapped between the two systems as well as colour information
- The user can filter data to optimize the process
- If you wish to visualise and interrogate the CATIA V5 or Creo data this can be done by using the integrated User Interface, which is included with CADverter
- 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 Creo and CATIA V5 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 CADverter for CATIA V5 - Creo product. For information regarding any of Theorem's product ranges please contact sales@theorem.com

Getting Started

Documentation

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 Catia V5 <> Creo 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

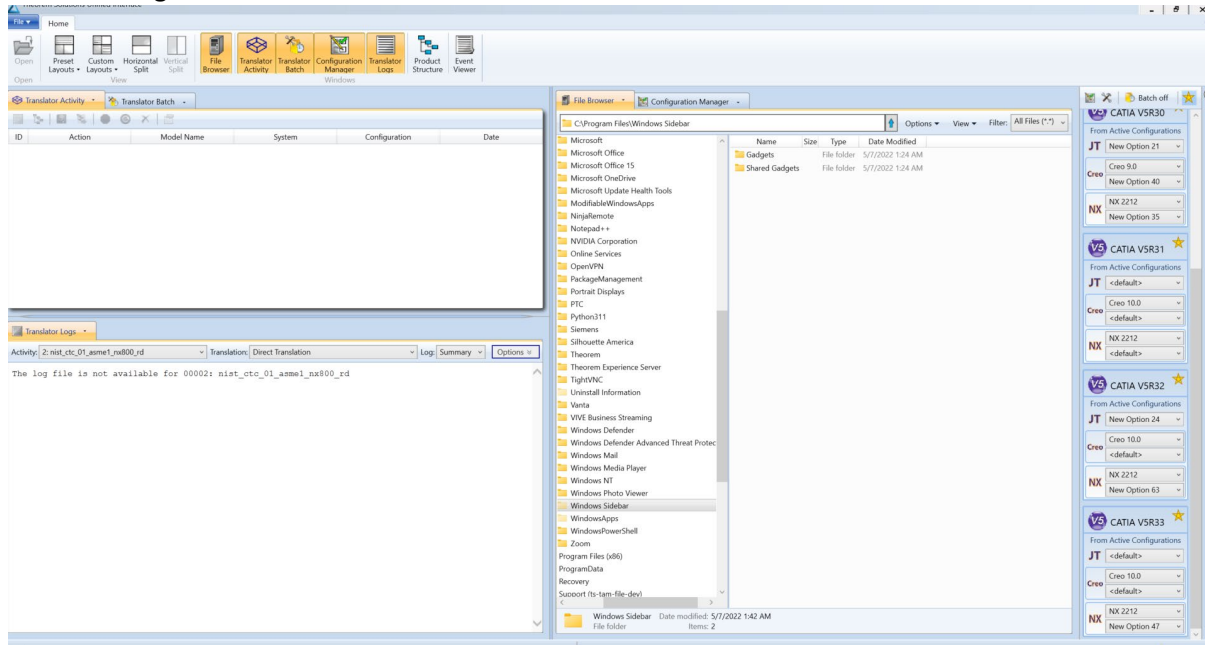
Default Translation – via the Unified Interface

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

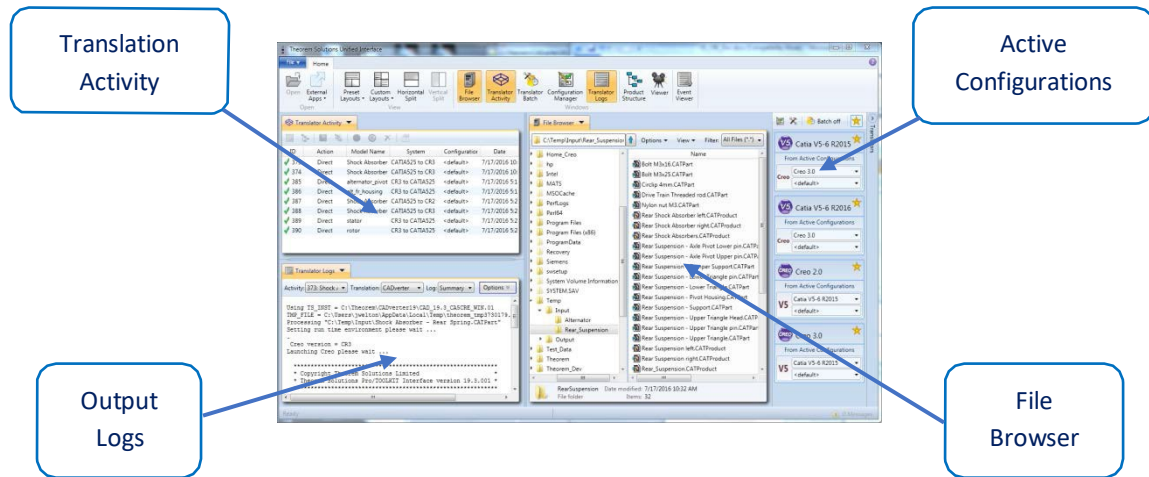
Alternatively, the Unified Interface can be run via a Windows Explorer selection in:

<UI_installation_directory>\bin\Unified_Interface.cmd

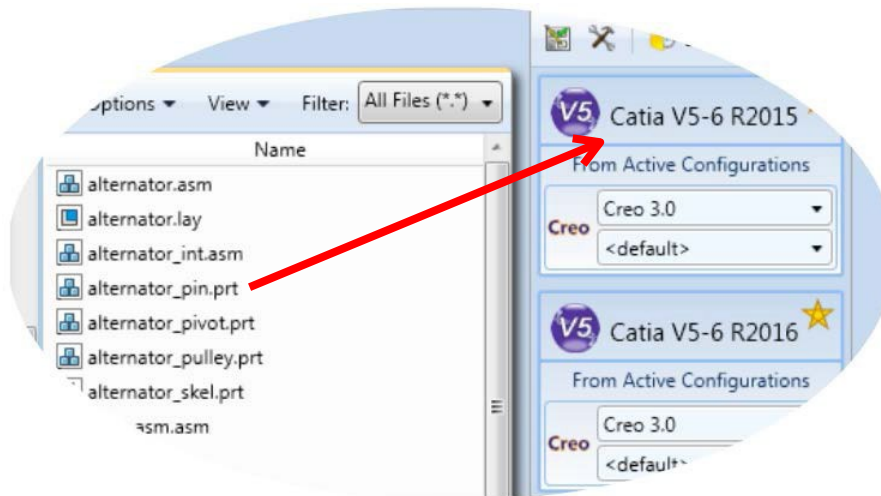
The following interface will be launched:



The default layout is split into 4 primary areas, which can be altered to the users prefer:

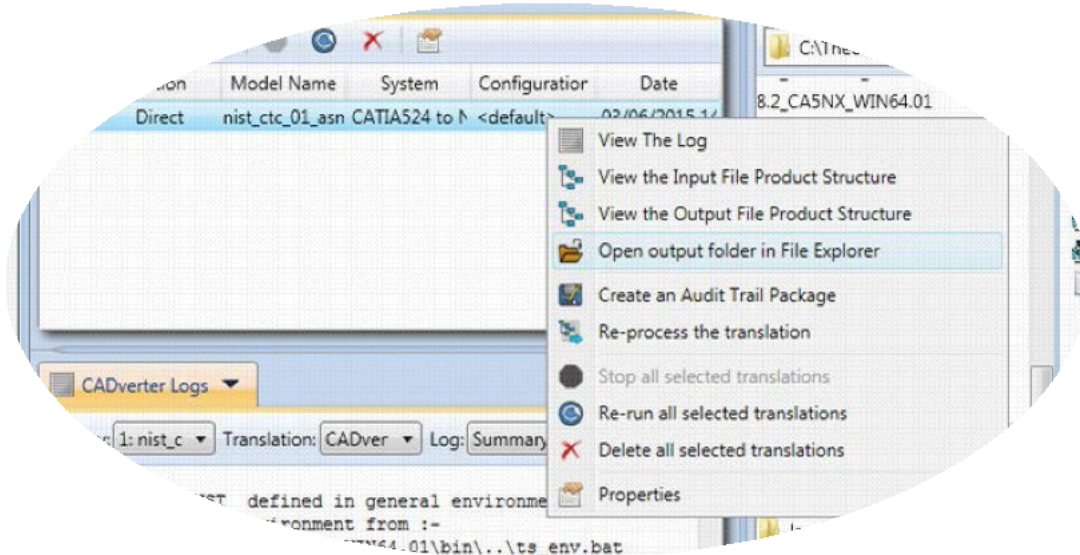


The simplest way to translate from CATIA V5 or CREO is to drag a file from the file Browser Pane on to the Active Configurations for the translation you require.



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

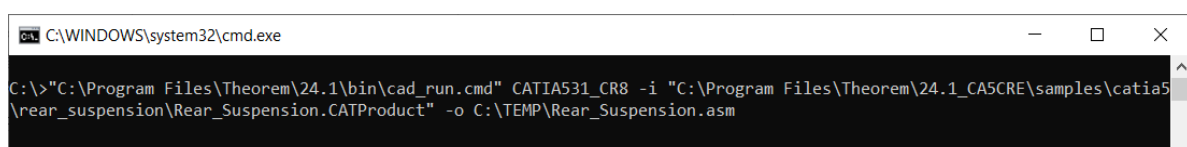
Running a translation via the command line can be carried out via the **cad_run.cmd** file located in the **<installation_directory>\bin** directory. The format of the command is as follows when translating from CATIA V5 to CREO:

**<Translator_installation_directory>\bin\cad_run.cmd CATIA5[XX]_CR[Y]
<input_file> <output_file>**

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

**<Translator_installation_directory>\bin\cad_run.cmd CR[Y]_CATIA5[XX]
<input_file> <output_file>**

Note! Replace the [XX] seen in the example with the version of CATIA V5 you are using and replace the [Y] with the version of Creo you are using. E.g. for CATIA V5-6R2020, change to CATIA530 and for CREO 6 change to CR6:



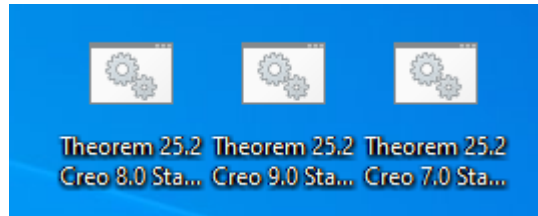
```
C:\>"C:\Program Files\Theorem\24.1\bin\cad_run.cmd" CATIA531_CR8 -i "C:\Program Files\Theorem\24.1_CA5CRE\samples\catia5\rear_suspension\Rear_Suspension.CATProduct" -o C:\TEMP\Rear_Suspension.asm
```

The example above will translate a CATIA V5 sample file provided within the installation and produce the following output to the target location. In this case:

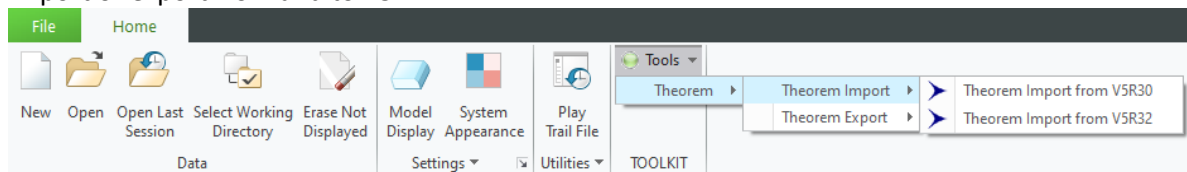
C:\Temp\Rear_Suspension.asm

Default Translation – via the Interactive method

It is possible to run translations via the interactive method for V5<>Creo using the Theorem menu within Creo. To do this launch the start up commands which will be on your desktop or install directory.



The next step would be once this is open within the main ribbon under Tools you will be able to Import or export from and to V5.



CADverter Customization

CADverter 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 CREO

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

- **CATIA V5 Read** – Those arguments that affect how data is read from CATIA V5
- **CREO Write** – Those arguments that affect how the data is written to CREO
- **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	Creo Write	Entity Mask	General
Option Name	Value		
Retain Assembly Structure	<input checked="" type="checkbox"/>		
Read PMI	<input type="checkbox"/>		
PMI Level	All		
Read Captures	<input checked="" type="checkbox"/>		
Read FTA Reference Geometry	<input type="checkbox"/>		
Maintain CATIA V5 Instance Names	<input type="checkbox"/>		
Read Face Colours	<input type="checkbox"/>		
Read Face Opacity	<input type="checkbox"/>		




Each of these options is described below:

Option	Description
Retain Assembly Structure	Maintain assembly structure in target file (<i>Default is ON</i>) <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>offditto (no structure, single part created)</i>
Read PMI	Enables PMI data read from the V5 file. (<i>Default is OFF</i>). <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>dont_read_pmi – to turn off</i>
PMI Level	A secondary argument to ‘Read PMI’ and allows control of the level of PMI to be read. (<i>Default is ALL when ‘Read PMI’ is marked as ON.</i>) <ul style="list-style-type: none"> • Options Available (command line syntax in italics and square brackets next to the option) <ul style="list-style-type: none"> ○ All - [read_pmi] ○ Part Level - [read_part_pmi] ○ Assembly Level - [read_assy_pmi] ○ Assembly Set (From CATPart) - [read_part_assy_pmi]
• Read Captures	<ul style="list-style-type: none"> • A secondary argument to ‘Read PMI’ and allows the control over whether captures are read as part of the process. (<i>Default is OFF</i>) when ‘Read PMI’ is marked as ON. <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ read_captures – to turn on ○ dont_read_captures – to turn off
• Read FTA Reference Geometry	<ul style="list-style-type: none"> • Enables reading of FTA Reference Geometry (<i>Default is OFF</i>) <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ read_geometry – to turn on

<ul style="list-style-type: none"> • Maintain CATIA V5 Instance Names 	<ul style="list-style-type: none"> • Honours CATIA V5 <i>Tools->Options->Infrastructure->Product Structure->Nodes Customization</i> panel settings (Default is OFF)  <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ ditto_naming_v5 – to turn on
<ul style="list-style-type: none"> • Read Face Colours 	<ul style="list-style-type: none"> • Read the face colours (Default is Off) • Command Line Syntax: <ul style="list-style-type: none"> ○ read_face_colours – to turn on
<ul style="list-style-type: none"> • Read Face Opacity 	<ul style="list-style-type: none"> • Read the opacity of the face (Default is Off) • Command Line Syntax: <ul style="list-style-type: none"> ○ read_materials – to turn on

CREO Write Arguments

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

CATIA V5 Read	Creo Write	Entity Mask	General
Option Name	Value		
Simplify Geometry	<input type="checkbox"/>		
Use existing Assembly Parts	<input checked="" type="checkbox"/>		
Trim part name	Off		
Seed Part	<input type="text"/>		
Seed Assembly	<input type="text"/>		
Attribute Mapping file	<input type="text"/>		

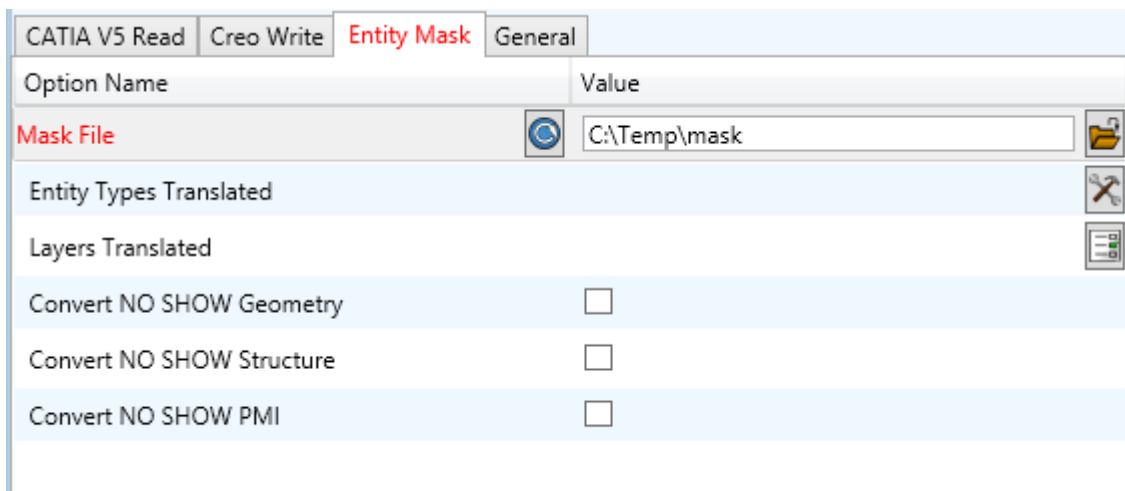
Each of these options is described below:

Option	Description
Simplify Geometry	Attempt to write Analytical Geometry where possible (Default is Off). <ul style="list-style-type: none"> • Command Line Syntax

	<ul style="list-style-type: none"> ○ <i>simplify</i>
Use existing Assembly Parts	<p>If the output Folder already contains output Files of the same name, do not Overwrite those Files (<i>Default is On</i>).</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>use_parts</i>
Trim part name	<p>If the part name is too long, use the first X characters and the final (30 - X) characters to reach the 30 character limit. (<i>Default is off</i>).</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>chop_name X</i>
Seed Part	<p>Use a Creo Seed Part file when creating the Creo output</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>seed_prt</i>
Seed Assembly	<p>Use a Creo Seed Assembly file when creating the Creo output</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>seed_asm</i>
Attribute Mapping File	<p>Select a standard property mapping file</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>cad_prop_map_file</i>

CATIA V5 to CREO Entity Masking Arguments

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



Each of these options is described below:

Option	Description
--------	-------------

Mask File	<p>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:</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>Mask <path\filename></i>
Entity Types Translated	<p>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"</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>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</i>
Layers Translated	<p>Specifies a selection list from which to select which layers are to be processed.</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>A single entry of ON ALL LAY Must precede any Layer Mask command.</i> ○ <i>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</i> e.g.: OFF LAY 114,149,166,167,168
Convert NO SHOW Geometry	<p>Enables Hidden geometry to be processed (<i>Default is Off</i>)</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>Add the following entry to the Mask file</i> ON NOSHOW
Convert NO SHOW Structure	<p>Enables Hidden Assembly Structure to be processed (<i>Default is Off</i>)</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>Add the following entry to the Mask file</i> ON NOSHOW STR
Convert NO SHOW PMI	<p>Enables Hidden PMI to be processed (<i>Default is Off</i>)</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <i>Add the following entry to the Mask file</i> ON NOSHOW PMI

CATIA V5 to CREO General Arguments

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

Option Name	Value
Mass Properties	<input type="checkbox"/>
Advanced	<input type="text"/>

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 to the PDF document. Default is OFF. <ul style="list-style-type: none"> ○ Command Line Syntax: <ul style="list-style-type: none"> ▪ <i>mprops</i>
Advanced	Allows any of the Command Line Advanced arguments documented below to be passed to the Unified Interface invocation

Common Options for CREO to CATIA V5

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

- CREO Read – Those arguments that affect how data is read from CATIA V5
- CATIA V5 Write – Those arguments that affect how the data is written to CREO
- General – Those arguments that are common to ALL Publishing activities regardless of source data

CREO Read Arguments

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

Option Name	Value
Transfer Solids	<input checked="" type="checkbox"/>
Transfer Quilts	<input checked="" type="checkbox"/>
Transfer Datum Curves	<input checked="" type="checkbox"/>
Transfer Datum Surfaces	<input checked="" type="checkbox"/>
Read PMI	<input type="checkbox"/>
Fill PMI Text	<input type="checkbox"/>
Read Cables	<input checked="" type="checkbox"/>
Process SimpRep (Interactive only)	<input type="checkbox"/>
Create Empty Nodes	<input type="checkbox"/>
Read Sub Assembly Colours	<input type="checkbox"/>
Read Part Colours	<input checked="" type="checkbox"/>
Read Surface Colours	<input checked="" type="checkbox"/>
Instance Processing	Off
Instance name	<input type="text"/>

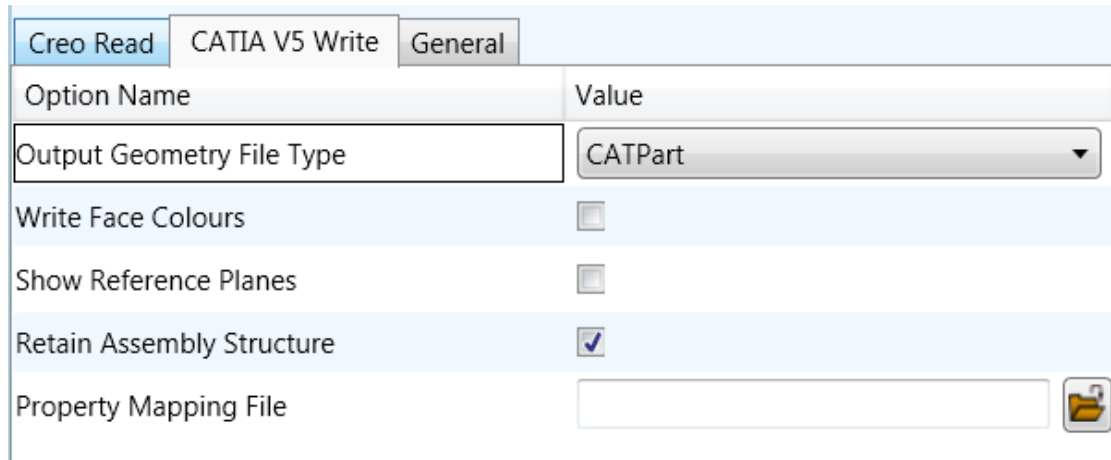
Each of these options is described below.

Option	Description
Transfer Solids	Enables solid processing. (Default is on). <ul style="list-style-type: none"> Command Line Syntax: <ul style="list-style-type: none"> no_solids – to Disable
Transfer Quilts	Enables quilt processing. (Default is on). <ul style="list-style-type: none"> Command Line Syntax: <ul style="list-style-type: none"> no_quilts – to Disable
Transfer Datum Curves	Enables Datum Curve processing. (Default is on). <ul style="list-style-type: none"> Command Line Syntax: <ul style="list-style-type: none"> no_datum_curves – to Disable
Transfer Datum Surfaces	Enables Datum Surface processing. (Default is on). <ul style="list-style-type: none"> Command Line Syntax: <ul style="list-style-type: none"> no_datum_surfaces – to Disable
Read PMI	Enables Datum Surface processing. (Default is off). <ul style="list-style-type: none"> Command Line Syntax: <ul style="list-style-type: none"> read_pmi
Fill PMI Text	Enabled when 'Read PMI' is selected. Improves the quality of PMI, but increases output size and processing time. (Default is off) <ul style="list-style-type: none"> Command Line Syntax: <ul style="list-style-type: none"> fill_pmi_text

Read Cables	<p>Enables the reading of Cable data from Creo. (Default is off). Note that in the Creo Configuration Editor, the setting <code>display_thick_cables</code> should be set to yes.</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <code>read_cables</code>
Process Simprep	<p>Enables the reading of a specified Simplified Representation. This is only available when processing data interactively and using the option via the Configuration Manager.</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <code>process_simprep</code>
Create Empty Nodes	<p>If a part is missing, create an empty leaf node for it. (Default is off)</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <code>create_empty_part</code>
Read Sub Assembly Colours	<p>Read colour information set on sub-assembly level (Default is off)</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <code>sub_assy_colours</code> ○ <code>sub_assy_colours_off</code> (default)
Read Part Colours	<p>Read colour information set on parts. (Default is on)</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <code>part_colours</code> (default) ○ <code>part_colours_off</code>
Read Surface Colours	<p>Read colour information set on surfaces. (Default is on)</p> <ul style="list-style-type: none"> • Command Line Syntax: <ul style="list-style-type: none"> ○ <code>surface_colours</code> (default) ○ <code>surface_colours_off</code>
Instance Processing	<p>Process a defined instance. Contains 3 options: (Not to be used in conjunction with Process Specified Instance). (Default is off).</p> <ul style="list-style-type: none"> • Off • List all instances in log file - List all instances of Family table to the progress file. <ul style="list-style-type: none"> ○ <i>Command Line Syntax</i> <ul style="list-style-type: none"> ▪ <code>instance LIST_ALL</code> • <i>Process Specified Instance – Only process the specified instance. The Text Box Instance Name will become active.</i> <ul style="list-style-type: none"> ○ <i>Command Line Syntax</i> ○ <code>instance [instance_name]</code>
Instance Name	<p>Enter the instance name to process. <i>Only activates when 'Process Specified Instance' selected.</i></p>

CATIA V5 Write Arguments

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



Each of these options is described below:

Option	Description
Output Geometry File Type	<p>Output Geometry file type (Default is CATPart - a CATProduct will be created for an assembly)</p> <p>Command Line Syntax</p> <ul style="list-style-type: none"> ○ <i>output_type CATPart (CATPart)</i> ○ <i>output_type model (Model)</i> ○ <i>output_type cgr (Cgr)</i> ○ <i>output_type igs (Igs)</i> ○ <i>output_type CATShape (CATShape)</i> ○ <i>create_cgr (Tessellated)</i>
Write Face Colours	<p>Processes Colours as applied to individual faces (Default is OFF)</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>face_colour (default)</i> ○ <i>solid_colour (use colour as applied to solids)</i>
Show Reference Planes	<p>Creates reference planes. Default is OFF.</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>Show: dont_blank_planes</i>
Retain Assembly Structure	<p>Maintain assembly structure in derive output (Default is ON).</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>off_ditto</i>
Property Mapping File	<p>Assign a property mapping file</p> <ul style="list-style-type: none"> • Command Line Syntax <ul style="list-style-type: none"> ○ <i>cad_prop_map_file [Path to file]</i>

CREO to CATIA V5 General Arguments

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

Creo Read	CATIA V5 Write	General
Option Name	Value	
Mass Properties	<input type="checkbox"/>	
Advanced	<input type="text"/>	

The option is described below:

Option	Description
--------	-------------

Mass Properties	<p>Allows Mass Property information to be read from the source data and written as attributes to the PDF document (<i>Default is off</i>)</p> <ul style="list-style-type: none"> ○ Command Line Syntax:
Advanced	<p>Allows any of the Command Line Advanced arguments documented below to be passed to the Unified Interface invocation</p>

Command Line Advanced Arguments

Any of the Advanced arguments can be added to the Command Line Invocation or to the General > Advanced field when run from within the User Interface.

Creo Arguments

Creo Argument	Description
pmi_pcurves	Store non planar PMI graphics (leaders not in the plane of the annotation)
unique_occ	Read multiple occurrences, eg support for occurrence PMI associations
read_assy_pmi	reads PMI in lower level assembly parts
ignore_std_views	Disable reading views with standard names (TOP, LEFT, etc).
ignore_view_list <file>	supply a list of view names to be ignored
no_exploded_views	Disables exploded views.
part_level_views	<i>Default: off</i> Enable the processing of Part Level Views within an assembly.
part_level_views_moved	Process part level views and move into assembly space, such that only one part instance's views are displayed.
part_level_pmi	<i>Default: off</i> Enable the processing of PMI on parts within an assembly.
view_part_name	Uses the part name in the view names to help identify the views when selected in PDF
views_geom_exploded <on off hybrid hybrid2>	<p><i>Default: hybrid</i></p> <p>Geometry grouped into assembly nodes for views is by default done on a view bases, which re-uses geometry where possible. This option creates a node for every item of geometry so that the views can hide/show them in the views. This can (depending upon the data and views) reduce the resulting file size and shorten the translation time, BUT in some cases the PDF is slow to respond due to the VIEW limitations.</p> <p><i>exploded - off</i> - create a node per view with all geometry / wire etc for that view</p> <p><i>explode - on</i> - create nodes for very solid / wireframe etc that can be referenced by views</p> <p><i>hybrid (default)</i> - mixture of explode on/off - solids being exploded and wireframe grouped (best compromise)</p> <p><i>hybrind2</i> - as hybrid with points also grouped</p>

CATIA V5 Arguments

CATIA V5 Arguments	Description
<ul style="list-style-type: none"> • convert_curves • 	<ul style="list-style-type: none"> • Converts curves to NURBS form
<ul style="list-style-type: none"> • convert_surfaces • 	<ul style="list-style-type: none"> • Converts surfaces to NURBS form
<ul style="list-style-type: none"> • face_opacity • 	<ul style="list-style-type: none"> • Read face opacity
<ul style="list-style-type: none"> • no_face_colour 	<ul style="list-style-type: none"> • Sets the default to SOLID colours
<ul style="list-style-type: none"> • noshow • noshow_geom • noshow_struct • noshow_pmi 	<ul style="list-style-type: none"> • Reads hidden geometry / structure / pmi
<ul style="list-style-type: none"> • output_mbd 	<ul style="list-style-type: none"> • Allows sub-part specification tree information to be read and presented to 3D PDF as product structure information. This option also enables the '<i>part_level_views</i>' and '<i>part_level_pmi</i>' 3D PDF options
<ul style="list-style-type: none"> • Read_geometry_edges 	<ul style="list-style-type: none"> • CATIA V5 has a display mode that 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

Large Assembly Processing

Overview

Large Assembly Processing (LAP) is available via command line only. Theorem's LAP mechanism was designed to be a method to efficiently converting large assemblies into their target format.

This process differs from the standard CAD-to-CAD conversion method. The standard method opens the entire assembly into memory and converts the assembly into the target system. Whereas LAP employs a method to convert each assembly leaf node independently and then processes the top-level assembly using the already created component files.

LAP Advantages:

- Consumes fewer system resources, RAM and CPU, thus allowing for larger assemblies to be processed on the same hardware.
- Can be used as a troubleshooting mechanism as files which failed to convert can be re-translated with different options. This requires the assembly process to be run again to ensure the assembly contains these components.

LAP Disadvantages:

- The use of LAP comes with the overhead of loading the libraries and licensing to convert each file independently. For this reason, the user may find that this process takes more time to complete.

LAP Process Overview

The LAP process begins with the user writing a "Wrapper Script" This script is a windows batch file that calls the initial translation using the *assy_script* argument. When this script runs, it will output a number of Viewer files and a component script (batch file *.bat) to the specified output folder. More details of the component script are provided in the applicable section.

A Viewer file is a Theorem proprietary file used as a temporary container for passing CAD information from the source to the target system. These files are maintained through the translation process. This can be useful if the translation is stopped and needs to be restarted. Using the existing Viewer files, the translation will continue from where it was stopped.

The component script, created by the wrapper script, contains a single line translation for each part file with the required translator arguments. This script can also be used to map the Viewer file names to the specified output file.

Once the part files have been created, the assemblies will be processed. Using the part files that were created when the component script ran, the assemblies are built using the transforms from the source CAD file. Thus ensuring all component files are at their proper location.

CATIA V5 to Creo

Review the “LAP Process Overview” section before continuing. This section outlines process in detail.

The wrapper script will create three component batch scripts. The first is created to process CATIA V5 data to Viewer files. The name of this script is *input_filename.vwr.bat*. The second component script is to process the Viewer files to CREO. The name of this file is *input_filename.vwr_2PROE.bat*. The third script is *input_filename.vwr_MOD.bat* and is used during the update process, discuss later in this section.

Command Line Arguments

Argument	Description
assy_script	<i>Invokes the LAP process</i>
as_leaf_node_args	<i>Translator arguments can be supplied for each leaf node conversion. Provides a method to supply different translator arguments which apply only to the component part level. e.g.: as_leaf_node_args “convert_curves convert_surfaces”</i>

Example Wrapper Script:

Note: The ^ symbol denotes continue to next line. Shown here to make the script easier to read, else it is a single line call to the translator.

```
C:\Program Files\Theorem\24.1\bin\cad_run.cmd ^
CATIA530_CR6 ^
C:\Temp\Input\Rear_Suspension\Rear_Suspension.CATProduct ^
C:\Temp\Output\Rear_Suspension\Rear_Suspension.prt ^
read_pmi_1 assy_script ^
as_leaf_node_args "read_pmi_1 dont_fill_pmi_text
dont_fill_pmi_arrows"
```

Updating the Assembly

The following methods can be used to update an already translated assembly into the same output folder. Any new files will be created. All Transforms will be re-processed, so component placement will be correct.

To begin, delete or rename the Creo files and the corresponding Viewer files to be re-translated. Set the environment variable `TS_REM_OUT_EXISTING_PARTS=1` either in the command prompt window used to run LAP or in the wrapper script. Then run the wrapper script. For this process, the *input_filename.vwr_MOD.bat* will be used to recreate the required Viewer files. Also, the *input_filename.vwr_2PROE.bat* will be recreated to only translate those new Viewer files to CREO Part files.

When the translation is complete, the required component files have been recreated as well as the assembly files. The assembly files are processed to ensure any modification to the component transforms are honored.

Troubleshooting Files

One advantage to LAP is that each output file creates its own log file. This may make it easier to troubleshoot that individual file if it is found that it did not translate properly. The user must decide what arguments are to be added and if they are CATIA read or CREO write arguments, or both.

CATIA Read Arguments Only – Begin by deleting the Viewer file and the CREO part file. Edit the Wrapper script to apply new arguments, and then run the translation.

CREO Write Arguments Only – Begin by deleting or renaming the CREO file in the output folder. The *input_filename.vwr_2PROE.bat* file can be edited to comment out each part file which does not require a new translation. Then edit the translator options for those files to be processed again. Run the *input_filename.vwr_2PROE.bat* script.

Alternatively, the required component translation can be copied and pasted into a command prompt window. The translator arguments edited in this window and the translation processed.

Both CATIA Read and CREO Write Arguments - Begin by deleting the Viewer file and the CREO part file. Edit the Wrapper script to apply new arguments. Be sure to use the *'as_leaf_node_args'* to apply options, where required, to only the component part files. Then run the wrapper script to begin the translation.

Creo to CATIA V5

Review the “LAP Process Overview” section before continuing. This section outlines process in detail.

The wrapper script will create the component batch script. This script will contain the command line translation for all required part files.

Command Line Arguments

Argument	Description
assy_script	<i>Invokes the LAP process</i>
as_leaf_node_args	<i>Translator arguments can be supplied for each leaf node conversion. Provides a method to supply different translator arguments which apply only to the component part level. e.g.: as_leaf_node_args “convert_curves convert_surfaces”</i>

Example Wrapper Script:

Note: The ^ symbol denotes continue to next line. Shown here to make the script easier to read, else it is a single line call to the translator.

```
C:\Program Files\Theorem\24.1\bin\cad_run.cmd ^
CR6_CATIA530 ^
C:\Temp\Input\Alternator\alternator.asm ^
C:\Temp\Output\Alternator\Alternator.CATProduct ^
no_datum_curves no_quilts no_datum_surfaces assy_script ^
as_leaf_node_args " surf_to_nurbs edge_to_nurbs "
```

If the translation is to be re-processed, the existing Viewer files will be used. To re-process the translation while overwriting the existing Viewer files, the following environment variable must be set:

```
TS_NO_USE_ASSY_TEMP_FILES=1
```

Updating the Assembly

The following methods can be used to update an already translated assembly into the same output folder. Any new files will be created. All Transforms will be re-processed, so component placement will be correct.

To begin, delete or rename the CATIA files to be re-translated. Either use the environment variable listed previously to overwrite the exiting Viewer file, or delete the Viewer file prior to running the translation. If unsure which viewer file to remove, the Component Batch script will map each Viewer file to an output CATIA filename. Then run the translation again.

Troubleshooting Files

One advantage to LAP is that each output file creates its own log file. This may make it easier to troubleshoot that individual file if it is found that it did not translate properly. The user must

decide what arguments are to be added and if they are CREO read or CATIA write arguments, or both.

CREO Read Arguments Only – Begin by deleting the Viewer file or setting the environment variable mention previously. Edit the Wrapper script to apply new arguments, and then run the translation.

CATIA Write Arguments Only – Find the file to be re-translated in the Component script. Edit the translator arguments. Either comment out all other lines (add REM to the beginning of the line) or copy the translation call and paste it into a command prompt window. Run the translation.

Both Read and Write Arguments – Begin by deleting the Viewer file or setting the environment variable mention previously. Edit the Wrapper script. Be sure to add the required arguments to the `as_leaf_node_args` portion. Run the translation.

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.

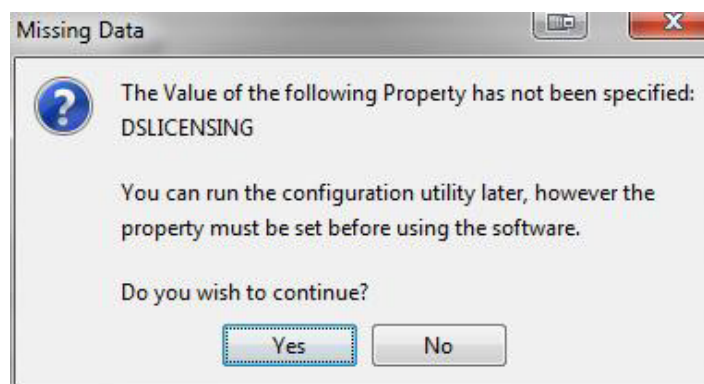
CATIA V5 Installation Directory

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.

For CATIA V5 R22 and lower releases, a choice is available of using the older LUM Legacy licensing system or the new Dassault Licensing. To enable this enter LEGACY in the DSLICENSING option.

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:

```
%TS_INST%\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 XP:

```
C:\Documents and Settings\All Users\Application Data\DassaultSystemes\CATEnv
```

Windows 7, 8 & 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 CADverter 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 attempt 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.V5R26.B26
```

Note: The extension **.txt** is not required. The user can specify a release specific name using

CATIAV5R<XX>_ENV e.g. set CATIAV5R25_ENV=CATIA.V5R26.B26

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.




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