CADverter for INVENTOR - JT

Product Release Version 24.2

USER GUIDE

Document Revision: 1.0

Issued: 19/10/2021

Contents

Contents

Overview of CADverter

About Theorem

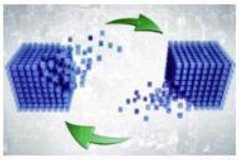
Theorem Solutions is a world leader in the field of Engineering Data Services and Solutions. This leadership position stems from the quality of our technology and the people in the company. Quality comes not only from the skills and commitment of our staff, but also from the vigorous industrial use of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

We are proud that the vast majority of the world's leading Automotive, Aerospace, Defence, 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.

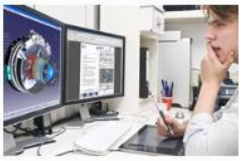
What is CADverter?

CADverter is one of 5 core Theorem brands which consist of:



CADverter

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



Multi-CAD

Interactive integration of non-native 3D data formats into the native CAD system

Visualize 3D

Direct translation of 3D data for the purpose of Visualization

Publish 3D

The creation of documents enriched with 3D content

Process Automation

Applications to automate any Data Exchange and collaboration processes

The INVENTOR - JT Translator

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

Theorem's CADverter product for Inventor to JT is a direct converter between Inventor assemblies (.iam files) and part (.ipt files) and Siemens JT parts. It enables the user to convert all forms of mechanical design geometry, as well as assembly and attribute information, between these two systems.

The Inventor-JT CADverter can be used interactively or in a batch mode, from a standard GUI Interface, offering combined viewing, data filtering and translation capabilities.

Primary Product Features

- Converts all types of geometry, wire frame, surfaces, trimmed surfaces (faces) and solid models
- Converts assembly structure between the systems
- Converts attribute data including colour and layer information
- The conversion process can be run Interactively or in Batch mode
- Data can be filtered by layer and entity type
- Geometry can be filtered and selectively processed

Primary Product benefits?

- Direct conversion between Inventor 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 CADverter for Inventor-JT product. For information regarding any of Theorem's product ranges please contact sales@theorem.com

Getting Started

Documentation

The latest copy of this 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.

Installation Media

The latest copy of Theorem software can be found via our web site at:

http://www.theorem.com/Product-Release-Notes

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 CD or ZIP file download provided.

Currently, there are 2 distinct installation stages that are required.

To install the translator, select the ***Setup.exe*** file and follow the installation process. For a full guide to the process, please see our 'Translator Installation Process' demonstration video located here.

In addition, the Theorem Unified Interface will also need to be installed. The installation process is the same as for the Translator. For a full guide to the process, please see our 'Translator Installation Process' demonstration video located here.

License Configuration

In order for the translation to run successfully, the Theorem license file provided to you needs to be configured using FlexLM. For a full guide to this process, please see our 'FlexLM License Set Up and Configuration' demonstration video located here.

Running the Product

Once configured and licensed, the product is ready to be run.

There are 2 distinct ways of running the translator:

- Via the Theorem Unified Interface
    - The Unified Interface offers a Desktop Environment that allows CAD and Visualization data to be viewed pre and post translation.
- Via the Command Line
    - The Command Line Interface provides a direct method of invoking the translator. It can be used via a command shell or called via a third party application as part of a wider process requirement.

Using the Product

Default Translations

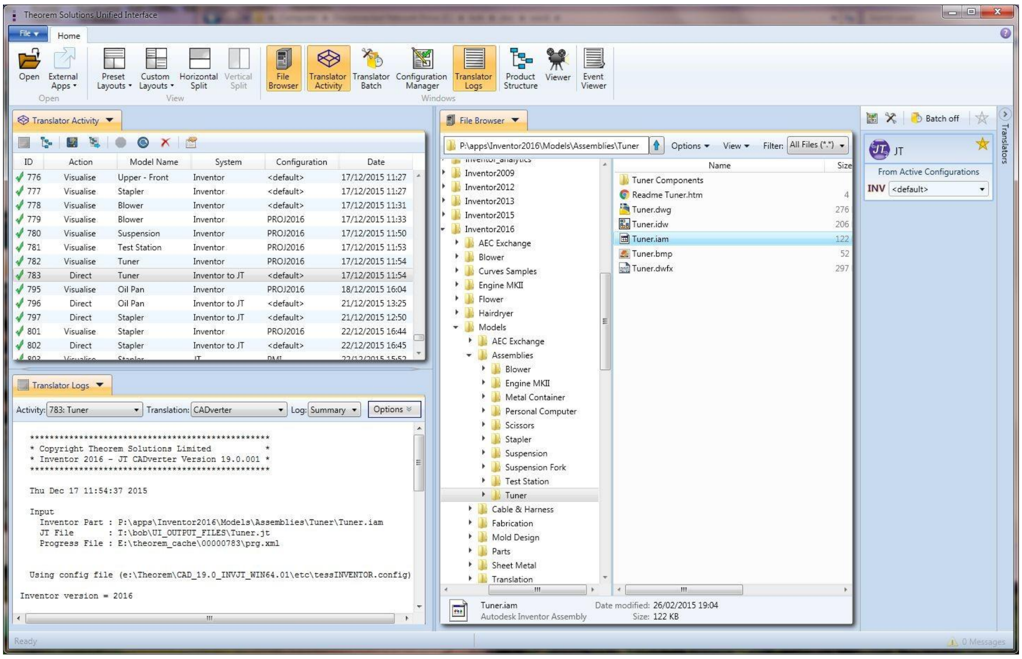
Default Translation – via the Unified Interface

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:



ID	Action	Model Name	System	Configuration	Date
✓ 776	Visualise	Upper - Front	Inventor	<default>	17/12/2015 11:27
✓ 777	Visualise	Stapler	Inventor	<default>	17/12/2015 11:27
✓ 778	Visualise	Blower	Inventor	<default>	17/12/2015 11:31
✓ 779	Visualise	Blower	Inventor	PROJ2016	17/12/2015 11:33
✓ 780	Visualise	Suspension	Inventor	PROJ2016	17/12/2015 11:50
✓ 781	Visualise	Test Station	Inventor	PROJ2016	17/12/2015 11:53
✓ 782	Visualise	Tuner	Inventor	PROJ2016	17/12/2015 11:54
✓ 783	Direct	Tuner	Inventor to JT	<default>	17/12/2015 11:54
✓ 795	Visualise	Oil Pan	Inventor	PROJ2016	18/12/2015 16:04
✓ 796	Direct	Oil Pan	Inventor to JT	<default>	21/12/2015 13:25
✓ 797	Direct	Stapler	Inventor to JT	<default>	21/12/2015 12:50
✓ 801	Visualise	Stapler	Inventor	PROJ2016	22/12/2015 16:44
✓ 802	Direct	Stapler	Inventor to JT	<default>	22/12/2015 16:45

Activity: 783: Tuner Translation: CADverter Log: Summary

```
 ***********************************************
 * Copyright Theorem Solutions Limited         *
 * Inventor 2016 - JT CADverter Version 19.0.001 *
 ***********************************************

 Thu Dec 17 11:54:37 2015

 Input
   Inventor Part : P:\apps\Inventor2016\Models\Assemblies\Tuner\Tuner.iam
   JT File       : T:\bob\UI_OUTPUT_FILES\Tuner.jt
   Progress File : E:\theorem_cache\00000783\prg.xml


 Using config file (e:\Theorem\CAD_19.0_INVJT_WIN64.01\etc\tessINVENTOR.config)

Inventor version = 2016
```

P:\apps\Inventor2016\Models\Assemblies\Tuner Filter: All Files (*.*)

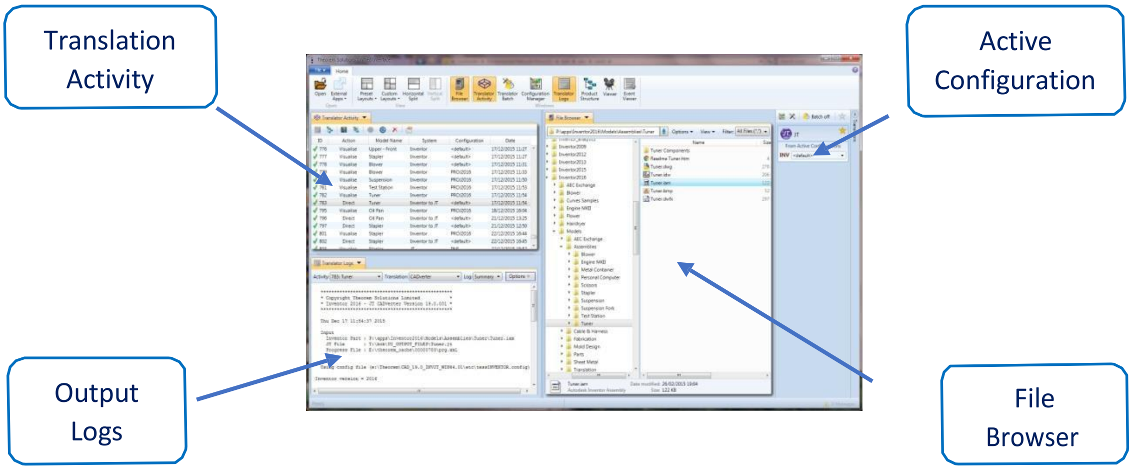
```
Inventor2009
Inventor2012
Inventor2013
Inventor2015
Inventor2016
  AEC Exchange
  Blower
  Curves Samples
  Engine MKII
  Flower
  Hairdryer
  Models
    AEC Exchange
    Assemblies
      Blower
      Engine MKII
      Metal Container
      Personal Computer
      Scissors
      Stapler
      Suspension
      Suspension Fork
      Test Station
      Tuner
    Cable & Harness
    Fabrication
    Mold Design
    Parts
    Sheet Metal
    Translation
```

Name	Size
Tuner Components	
Readme Tuner.htm	4
Tuner.dwg	276
Tuner.idw	206
Tuner.iam	122
Tuner.bmp	52
Tuner.dwfx	297

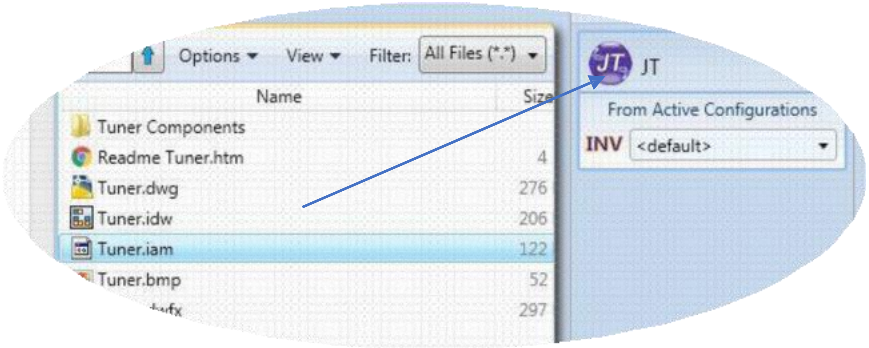
Tuner.iam Date modified: 26/02/2015 19:04
Autodesk Inventor Assembly Size: 122 KB

From Active Configurations
INV <default>

The default layout is split into 4 primary areas that can be altered to the user’s preference:



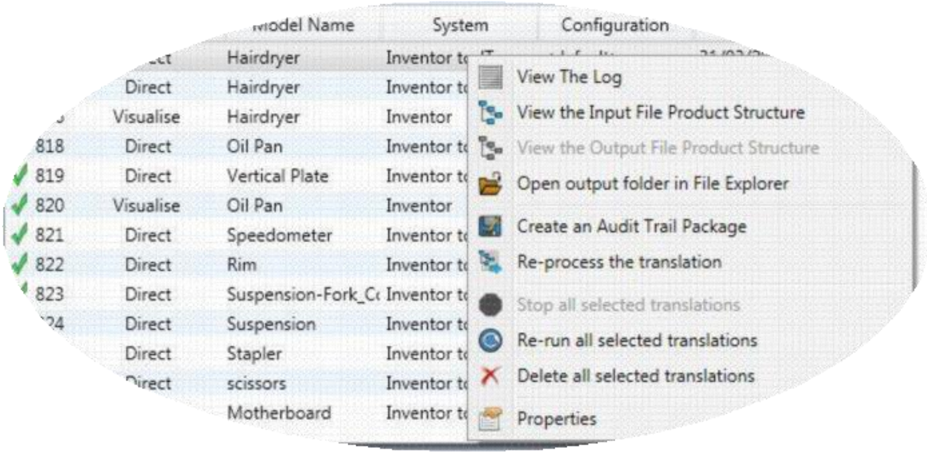
The simplest way to translate from INVENTOR is to drag a file from the file Browser Pane on to the Active Configurations for the translation you require.



Name	Size
Tuner Components	
Readme Tuner.htm	4
Tuner.dwg	276
Tuner.idw	206
Tuner.iam	122
Tuner.bmp	52
..dfx	297

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:

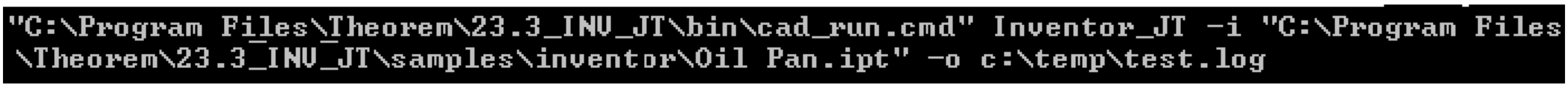

View The Log
View the Input File Product Structure
View the Output File Product Structure
Open output folder in File Explorer
Create an Audit Trail Package
Re-process the translation
Stop all selected translations
Re-run all selected translations
Delete all selected translations
Properties

		Model Name	System	Configuration
	..ct	Hairdryer	Inventor t	
	Direct	Hairdryer	Inventor t	
	Visualise	Hairdryer	Inventor	
818	Direct	Oil Pan	Inventor t	
✓ 819	Direct	Vertical Plate	Inventor t	
✓ 820	Visualise	Oil Pan	Inventor	
✓ 821	Direct	Speedometer	Inventor t	
✓ 822	Direct	Rim	Inventor t	
823	Direct	Suspension-Fork_Cc	Inventor t	
..24	Direct	Suspension	Inventor t	
	Direct	Stapler	Inventor t	
	..irect	scissors	Inventor t	
		Motherboard	Inventor t	

Default Translation – via the Command Line

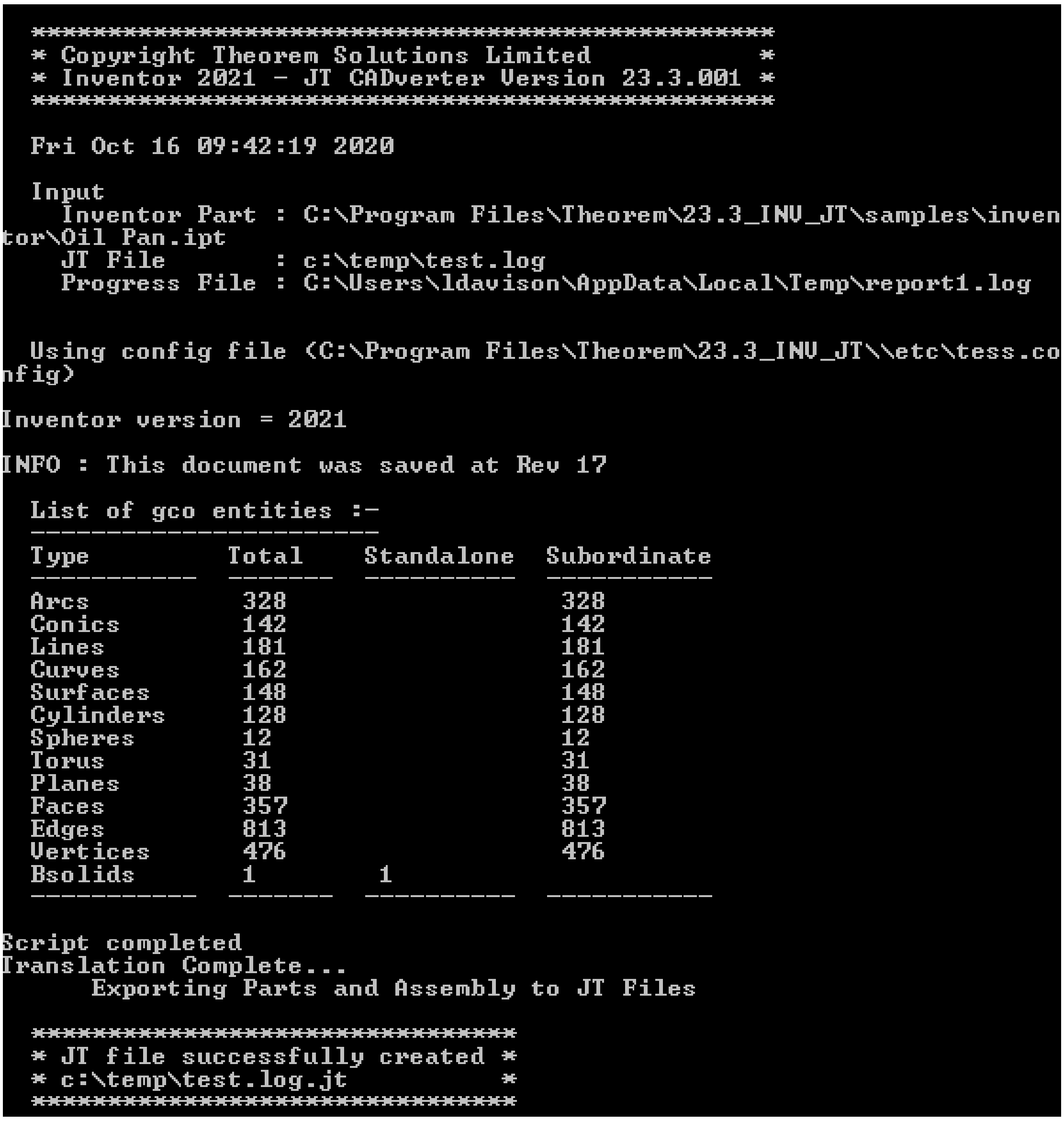
Running a translation via the command line can be carried out via the ***cad_run.cmd*** file located in the ***<installation_directory>\bin*** directory. The format of the command is as follows when translating from INVENTOR to JT:

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



```
"C:\Program Files\Theorem\23.3_INV_JT\bin\cad_run.cmd" Inventor_JT -i "C:\Program Files
\Theorem\23.3_INV_JT\samples\inventor\Oil Pan.ipt" -o c:\temp\test.log
```

The example above will translate an Inventor sample file provided within the installation and produce the following screen output:



```
  **************************************************
  * Copyright Theorem Solutions Limited            *
  * Inventor 2021 - JT CADverter Version 23.3.001 *
  **************************************************

  Fri Oct 16 09:42:19 2020

  Input
    Inventor Part : C:\Program Files\Theorem\23.3_INV_JT\samples\inven
tor\Oil Pan.ipt
    JT File       : c:\temp\test.log
    Progress File : C:\Users\ldavison\AppData\Local\Temp\report1.log


  Using config file (C:\Program Files\Theorem\23.3_INV_JT\\etc\tess.co
nfig)

Inventor version = 2021

INFO : This document was saved at Rev 17

  List of gco entities :-
  -----------------------
  Type          Total    Standalone  Subordinate
  -----------   -------  ----------  -----------
  Arcs            328                  328
  Conics          142                  142
  Lines           181                  181
  Curves          162                  162
  Surfaces        148                  148
  Cylinders       128                  128
  Spheres         12                   12
  Torus           31                   31
  Planes          38                   38
  Faces           357                  357
  Edges           813                  813
  Vertices        476                  476
  Bsolids         1         1
  -----------   -------  ----------  -----------

Script completed
Translation Complete...
      Exporting Parts and Assembly to JT Files

  *********************************
  * JT file successfully created *
  * c:\temp\test.log.jt           *
  *********************************
```

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

C:\temp\test.jt

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.

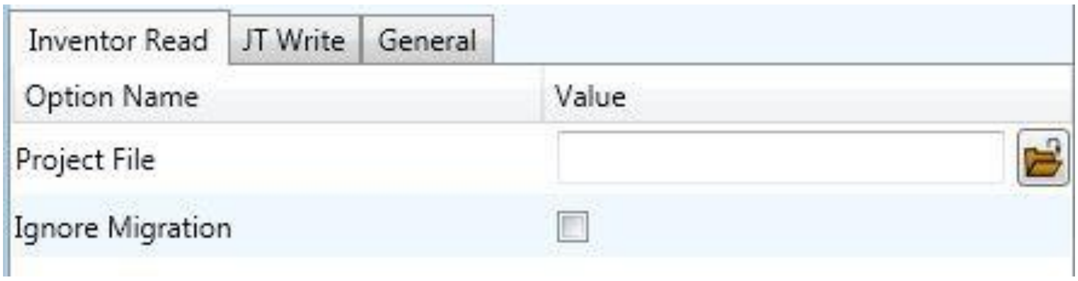
Common Options for INVENTOR to JT

Within the Configuration Manager pane of the Unified Interface, arguments that can be specified when publishing Inventor data into JT are grouped into 3 areas:

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

INVENTOR Read Arguments

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



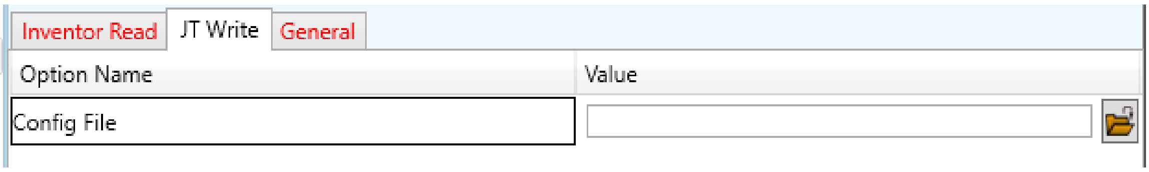
Option Name	Value
Project File	
Ignore Migration	□

Each of these options is described below:

Option	Description
Project File	This allows the user to specify an Inventor project file (*.ipg) which details search paths for parts within an assembly amongst other Inventor settings. ○ Command Line Syntax: ▪ *Project [file_name]*
Ignore Migration	This option allows the check for data migration to be omitted. Command Line Syntax: ▪ *ignore_migration*

JT Write Arguments

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



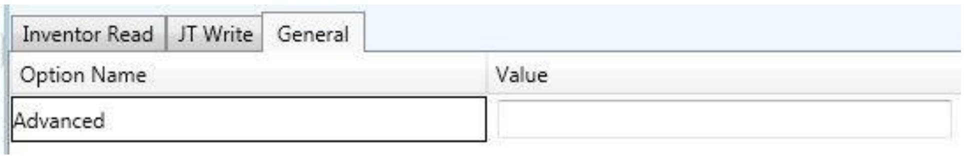
Option Name	Value
Config File	

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. Command Line Syntax • -z [path_to_file

INVENTOR to JT General Arguments

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



Option Name	Value
Advanced	

Each of these options is described below:

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

INVENTOR to JT Advanced Arguments

Theorem's INVENTOR to JT translator has been configured with default settings that optimise 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.

The following table describes useful Advanced Arguments that can be entered into the General Tab -> Advanced field:

Option	Description
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 ○ Command Line Syntax ▪ *nopstolmodel – to turn off*
Factor	Specify the factor level of Parasolid Tolerant Modelling when turned on. Default is 3. ○ 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
	○ Command Line Syntax ▪ *nosew – to turn off*
Tolerance	Specify the tolerance for the sew command above. Default is 0.01. ○ Command Line Syntax ▪ *pssew 0.01*
Incremental Sewing	Enables incremental sewing when used with Sew Parasolid Bodies. Default is ON. ○ 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. ○ 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. ○ 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. ○ 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. ○ 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. ○ 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. ○ 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. ○ 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. ○ Command Line Syntax ▪ *simplify_solids – to turn on*

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 config file will be used:-

> **tessINVENTOR.config** in ***%TS_INST%\etc*** directory (where TS_INST = Installed directory)

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	includeBrep = false

	TRUE false 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
JtFileFormat	64,70,80,81,82,90, 91,92,93,94,95,100	JtFileFormat = “95”
includeULP	PASSTHROUGH TRUE FALSE	includeULP = “PASSTHROUGH”
ulpPrecision	Real Value	ulpPrecision = 0.001
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
Label	*any string*	“ud_FINE”
FeatureSuppression	*any integer*	FeatureSuppression = 0
Simplify	*any number*	Simplify = 0.60
AdvCompressionLevel	*any number*	AdvCompressionLevel = 0.0
ULP	true false	
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>"

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
InventorOptions	InventorOptions {	InventorOptions {	
ProjectFile	Full system file path	Windows example ProjectFile = "P:\apps\Inventor2009 \Samples.ipj"	“”
IgnoreMigration	true/TRUE false/FALSE	IgnoreMigration = true	false
ReportFilename	Full system file path	Windows example ReportFilename = P:\caddata\translation\ result\part55	Windows system C:%TEMP%\t scprogressyi
OutputUnits	mm millimetres cm centimetres m metre metres inches feet yards inputUnits	OutputUnits = mm	inputUnits
StructureOutputType	JT PLMXML PLMXMLJT	StructureOutputType = JT	JT
PLMXMLPropertyMappingFile	*File Name*	PLMXMLPropertyMappingFile = “mapping_file.txt”	“”

brepType	XT JT XTJT	brepType = XT	JT
ParasolidTolerantModelling	true/TRUE false/FALSE	ParasolidTolerantMo de lling = true	true
ParasolidTolerantModellingFactor	*Any positive integer*	ParasolidTolerantMo de llingFactor = 3	3
SewParasolidBodies	true/TRUE false/FALSE	SewParasolidBodies = true	true
SewParasolidBodiesTol	*Any number*	SewParasolidBodiesTol = 0.01	0.01
IncrementalSewing	true/TRUE false/FALSE	IncrementalSewing = true	true
IncrementalSewingNoOfIterations	true/TRUE false/FALSE	IncrementalSewingN oO fIterations = 5	5
ExplodeSolidstoFaces	true/TRUE false/FALSE	ExplodeSolidstoFaces = false	false
SplitDiscontinuousSurfaces	true/TRUE false/FALSE	SplitDiscontinuousSur fa ces = true	false
ForceBodyCreation	true/TRUE false/FALSE	ForceBodyCreation = true	true
FixDegenerateEdges	true/TRUE false/FALSE	FixDegenerateEdges = true	true
FaceEdgeTol	*Any number*	FaceEdgeTo l = 0.000006	0.000006
FixSmallFeaturesSolids	true/TRUE false/FALSE	FixSmallFeaturesSolids = false	false
FixSmallFeaturesOpenSolids	true/TRUE false/FALSE	FixSmallFeaturesOpe nS olids = false	false
SimplifyGeometry	true/TRUE false/FALSE	SimplifyGeometry = false	false
BrepWireframe	true/TRUE false/FALSE	BrepWireframe = true	true
ProduceTessellatedOutput	true/TRUE false/FALSE	ProduceTessellatedO ut put = false	false
ExpandPart	true/TRUE false/FALSE	ExpandPart = false	false
ReuseSolids	true/TRUE false/FALSE	ReuseSolids = false	false
CADPropertyMappingFile	*File Name*	CADPropertyMappin gFi le =	“”

SavedViewsViewSetName	*"string of characters"*	SavedViewsViewSetNa me	"SavedViews"
AnnotationPlanesViewSetName	*"string of characters"*	AnnotationPlanesView SetName	"AnnotationPlanes"
Close brace	}	}	

Appendix C – 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.

The format is as follows:

- Lines 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,,
```