



ASFALIS Components Administration Manual

ASFALIS Adapter & ASFALIS Optimizer

August 2020

Elysium Co. Ltd.

Index

1. Overview	1
1.1. What is ASFALIS	1
1.2. What is ASFALIS Optimizer	1
1.3. Components	1
1.4. About Product ID	2
2. Environment Settings	5
2.1. Files and Arguments to Control the Data Translation	5
2.2. Configuration File	5
2.2.1. Keyword Definitions	6
2.2.2. Reference Keywords and Use Environment Variables	15
2.2.3. Multiple Entries in Configuration File	15
2.2.4. Samples	16
2.3. Command Line Arguments	17
2.3.1. Command Line Arguments	17
2.3.2. Sample Command Line Arguments	18
2.4. XML Scenario	19
2.4.1. Format for the Scenarios	20
2.4.2. Execution Example	31
2.5. Parameter File Settings	33
2.6. ELYBATCH Error Code	34
2.6.1. Note	34
3. ASFALIS Component	35
3.1. Adapter	35
3.1.1. CATIA V5 to ENF / ENF to CATIA V5	36
3.1.2. 3DEXPERIENCE to ENF / ENF to 3DEXPERIENCE	39
3.1.3. CATIA V4 to ENF / ENF to CATIA V4	42
3.1.4. NX I-deas to ENF / ENF to NX I-deas	43
3.1.5. Creo Parametric to ENF / ENF to Creo Parametric	45
3.1.6. NX to ENF / ENF to NX	48
3.1.7. Parasolid to ENF / ENF to Parasolid	51
3.1.8. SOLIDWORKS to ENF / ENF to SOLIDWORKS	52
3.1.9. Inventor to ENF / ENF to Inventor	54
3.1.10. Creo Elements/Direct to ENF / ENF to Creo Elements/Direct	55

3.1.11. ACIS to ENF / ENF to ACIS	56
3.1.12. IGES to ENF / ENF to IGES	57
3.1.13. STEP to ENF / ENF to STEP	57
3.1.14. STEP AP242 BOM to ENF / ENF to STEP AP242 BOM	59
3.1.15. JT to ENF / ENF to JT	61
3.1.16. PLM XML to ENF / ENF to PLM XML	62
3.1.17. iCAD to ENF / ENF to iCAD	64
3.1.18. CADmeister to ENF / ENF to CADmeister (Standalone)	65
3.1.19. CATIA V5 to ENF / ENF to CATIA V5 (Standalone)	68
3.1.20. Creo Parametric to ENF (Standalone)	69
3.1.21. NX to ENF (Standalone)	69
3.1.22. ENF to STL	70
3.1.23. ENF to XVL	70
3.1.24. ENF to 3D PDF	71
3.1.25. 3DXML to ENF	72
3.2. Healer, etc.	73
3.2.1. TOGO / CADCEUS Healer	73
3.2.2. ENF Polygon	73
3.3. Optimizer	74
3.3.1. PDQChecker	74
3.3.2. Geometry Simplifier	78
3.3.3. CAD Validator	81
3.3.4. Attribute Editor	98
3.3.5. Assembly Editor	106
3.3.6. Interference Checker	113
3.3.7. Polygon Optimizer	114
3.3.8. ENF Editor	115
3.3.9. 3D PDF Editor	132
3.4. Other	133
3.4.1. 3D Annotation Translation	133
3.4.2. Attribute Translation	146
3.4.3. License Management	162

1. Overview

1.1. What is ASFALIS

ASFALIS is a total solution of interoperability of CAD/CAM/CAE. By using this you can translate CAD native file into ENF (Elysium Neutral File) and heal or check it, then convert the file into the other CAD system or itself.

1.2. What is ASFALIS Optimizer

ASFALIS Optimizer is the component to optimize ENF(Elysium Neutral File) data by executing the function such as Geometry Simplification and PDQ check etc.

1.3. Components

Following adapters are available.

ENF Writer	ENF Reader
CATIA V5(CAA) to ENF (*1)	ENF to CATIA V5(CAA) (*2)
3DEXPERIENCE to ENF	ENF to 3DEXPERIENCE
CATIA V4 to ENF	ENF to CATIA V4
NX I-deas to ENF	ENF to NX I-deas
Parasolid to ENF	ENF to Parasolid
NX to ENF	ENF to NX
Creo Parametric to ENF	ENF to Creo Parametric
Autodesk Inventor to ENF	ENF to Autodesk Inventor
Creo Elements/Direct to ENF	ENF to Creo Elements/Direct
ACIS to ENF	ENF to ACIS
STEP to ENF	ENF to STEP
STEP AP242 BOM to ENF	ENF to STEP AP242 BOM
IGES to ENF	ENF to IGES
SOLIDWORKS to ENF	ENF to SOLIDWORKS
JT to ENF	ENF to JT

ENF Writer	ENF Reader
PLM XML to ENF	ENF to PLM XML
iCAD to ENF	ENF to iCAD
CADmeister to ENF (Standalone)	ENF to CADmeister (Standalone)
CATIA V5 to ENF (Standalone)	ENF to CATIA V5 (Standalone)
Creo Parametric to ENF (Standalone)	ENF to XVL
NX to ENF (Standalone)	ENF to STL
3DXML to ENF	ENF to 3D PDF

- (*1) Please note that with this adapter, the CATIA V5 data will be first translated to CADdoctor-type format, then to ENF.
- (*2) Please note that with this adapter, the ENF data will be first translated to CADdoctor-type format, then to CATIA V5.

Following optimizers are available.

- PDQ Checker
- Geometry Simplifier
- CAD Validator
- Attribute Editor
- Assembly Editor
- Interference Checker
- Polygon Optimizer
- ENF Editor

1.4. About Product ID

Unique ID is assigned to each CAD format (file format) which will be used to determine the source CAD and the target CAD system by the combination of IDs.

Product ID always comes with 6 digit values. It specifies the translation path (from CAD A to CAD B) by the 4 digits surrounded by "1 (one)" at the beginning and the end.

CAD	CAD ID
ENF	100
ACIS	102

CAD	CAD ID
NX I-deas	103
Creo Parametric	104
Parasolid	105
CATIA V4	106
SOLIDWORKS	109
Creo Elements/Direct	112
NX	114
CATIA V5	117
CADCEUS, CADmeister	121
TOGO	123
XVL	130
Inventor	142
iCAD	149
3DXML	163
CADmeister (Standalone)	167
CATIA V5 (Standalone)	170
Creo Parametric (Standalone)	171
NX (Standalone)	172
STEP AP242 BOM	175
3D PDF	176
STL	181
JT	184
PLM XML	194
3DEXPERIENCE	195
STEP	198
IGES	199

Examples

- 106031 means the data translation from CATIA V4 to NX I-deas.
- 114001 means the data translation from NX to ENF.

2. Environment Settings

2.1. Files and Arguments to Control the Data Translation

Configuration file	This is a file to specify the information about the translation environment such as the components to use, path for the CAD installation folder, license check settings and so on.
Command line arguments	This is to specify the information which changes for each data translation by the arguments. This includes the information of the input file, output file, parameter file, working directory and so on. It is also possible to specify an XML scenario by an argument and specify the necessary information such as input / output files in the XML scenario.
XML Scenario	This is a file to specify the order of the components, the input and output etc. for each component.
Parameter File	This is a file to specify the parameters (detailed configuration for the components). Parameter file needs to be specified either by the argument or writing a path to it in XML scenario.

2.2. Configuration File

Configuration file specifies the information about the translation environment such as the components to use, path for the CAD installation folder, license check settings and so on. Elybatch, the control program of ASFALIS components, executes the components based on the information in this file.

When specifying, the parameter name specified in advance (hereinafter referred to as "keyword") and the actual value (path, port number, etc.) should be described as a pair. As shown in the following example, from the beginning of the configuration file to the line before the first "PRODUCT <number>" is called the "Common section", and under "PRODUCT <number>" is called the "Product section".


```
#Common Section
ESERVER          C:\Asfalıs
PORT_NUM         5093
LIC_SERVER       XXX
TESSELLATOR      $ESERVER\tools\win

#Product Section
PRODUCT          117051
PATH_FROM_TRANSLATOR C:\catiav5\B28\intel_a\code\bin
FROM_TRANSLATOR   $ESERVER\CaaExe\win\B28
ADEK_DIR          $ESERVER\tools\win
TO_TRANSLATOR     $ESERVER\ParasolExe\win
PARASOLID_SCHEMA  $ESERVER\ParasolExe\schema
BATCH_STATUS
```

Please refer to "2.2.1, [“Keyword Definitions”](#)" for each keyword.

Required items are: ESERVER, PORT_NUM, and LIC_SERVER in common section; PRODUCT and BATCH_STATUS in product section.

- Lines that begin with "#" are ignored as comment lines.
- It will not work if there is a whitespace character before or after the keyword or value.
- At least one whitespace character (space or tab) is required between the keyword and the value.
- Lines with only whitespace characters are ignored.
- Use ANSI code page for the character codes.
- Use LF or CR+LF for the line feed code.

2.2.1. Keyword Definitions

Common Settings

ESERVER	
Description	Specify the root folder where ASFALIS module is located.
Example	ESERVER C:\Asfalıs
PORT_NUM	
Description	The port number which the Elysium license server listens to This is normally port 5093
Example	PORT_NUM 5093
LIC_SERVER	

Description	The hostname of the Elysium license server
Example	LIC_SERVER XXXX
TESSELLATOR	
Description	Path to the common modules
Example	TESSELLATOR \$ESERVER\tools\win
INTRCHECK	
Description	Path to the modules of Interference Checker
Example	INTRCHECK \$ESERVER\tools\win\elyIntrCheck\IntrCheck
EXTEND_ERROR_INFO	
Description	An option setting whether to export extended error messages (0/1/2) 0: Don't export 1: Export (Elybatch returns the codes within 1000 to 1010.) 2: Export (Elybatch returns all kinds of return codes.)
Example	EXTEND_ERROR_INFO 2
SCENARIO_FILE	
Description	Path to Elybatch scenario file
Example	SCENARIO_FILE C:\Asfalis\scenario\scenario.xml
SCENARIO_LOG	
Description	Path to Elybatch scenario log
Example	SCENARIO_LOG C:\Asfalis\log\scenariolog.xml
SCENARIO_PROGRESS	
Description	An option setting whether to export output progress log (0/1) 0: Don't export 1: Export (The progress report will be written to the "progress" file.) Please note that this is effective only when executed Elybatch in the scenario.
Example	SCENARIO_PROGRESS 1
NUM_OF_PROCESSES	

Description	<p>Specify the maximum number of processes available for translation.</p> <p>When no value is specified, an appropriate value will be applied in accordance with the corresponding adapter.</p> <p>When zero (0) is specified, an available maximum number of processes will be applied (in accordance with the environment.)</p> <p>This option is available only when it satisfies the following conditions:</p> <ul style="list-style-type: none"> * ENF Reader of NX, CATIA V5, or JT. - LEGACY_MODE is not specified or 0 is specified for it. - Product section has a priority if it is specified. <p>* ENF Writer of PLM XML and STEP AP242 BOM.</p> <ul style="list-style-type: none"> - LEGACY_MODE is not specified or 0 is specified for it. - Product section has a priority if it is specified. <p>* CAD Validator</p>
Example	NUM_OF_PROCESSES 4
BATCH_TIME_OUT	
Description	<p>This is maximum limit of translation running time. If the translation has not finished during specified time, all related processes are stopped and killed.</p> <p>(Default: 360 minutes: 6 hours)</p>
Example	BATCH_TIME_OUT 120

Settings on Each Conversion

PRODUCT	
Description	Product ID
Example	PRODUCT 117001
FROM_CAD_START_CMD	
Description	<p>Path to CAD startup command used for ENF Writer</p> <p>This is mandatory for the ENF Writers which requires the CAD environment such as the translation from NX I-deas or Creo Parametric.</p>
Example	FROM_CAD_START_CMD C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
PATH_FROM_TRANSLATOR	
Description	<p>An additional library path to add to the environment variables path</p> <p>This may be required for ENF Writer which requires the CAD environment such as the translation from CATIA V5.</p>

Example	PATH_FROM_TRANSLATOR C:\Program Files\Dassault Systemes\B20\intel_a\code\bin (* No line break is required.)
FROM_TRANSLATOR	
Description	Path to the executable to read an input CAD file
Example	FROM_TRANSLATOR \$ESERVER\CaaExe\win\B20
ADEK_DIR	
Description	Path to the adeKernel.exe program
Example	ADEK_DIR \$ESERVER\tools\win
TO_TRANSLATOR	
Description	Path to the executable to write an output CAD file.
Example	TO_TRANSLATOR \$ESERVER\CaaExe\win\B20
TO_CAD_START_CMD	
Description	Path to CAD startup command used for ENF Reader This is mandatory for the ENF Readers which requires the CAD environment such as the translation from NX I-deas or Creo Parametric.
Example	TO_CAD_START_CMD C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
PATH_TO_TRANSLATOR	
Description	An additional library path to add to the environment variables path This may be required for ENF Reader which requires the CAD environment such as the translation from CATIA V5.
Example	PATH_TO_TRANSLATOR C:\Program Files\Dassault Systemes\B20\intel_a\code\bin (* No line break is required.)
BATCH_STATUS	
Description	Reserved keyword The keyword: Mandatory (needs to be always specified) The value: Not required
Example	BATCH_STATUS
NUT_CHECK_DIR	
Description	Directory for PDQ Checker module
Example	NUT_CHECK_DIR \$ESERVER\tools\win

LM_FROM_TRANSLATOR, LM_TO_TRANSLATOR

Description	Path to the directory for the executable named "lmutil" This will be used to check the CAD licenses controlled by Flexlm license. Please note that this needs to be specified in a combination with "LICENSE" and "FEATURE," either LM_FROM_LICENSE and LM_FROM_FEATURE or LM_TO_LICENSE and LM_TO_FEATURE.
Example	LM_FROM_TRANSLATOR C:\Program Files\PTC\Creo 4.0\F000\Common Files\x86e_win64\obj (* No line break is required.)

LM_FROM_LICENSE, LM_TO_LICENSE

Description	Port number and hostname of the Flexlm license server Multiple servers can be specified.
Example	LM_FROM_LICENSE 7788@hostname1;7788@hostname2

LM_FROM_FEATURE, LM_TO_FEATURE

Description	FEATURE that is used for CAD license Up to 10 features can be specified.
Example	LM_FROM_FEATURE solid_modeling;gateway

LM_TO_FEATURE_OPERATOR, LM_FROM_FEATURE_OPERATOR

Description	<p>The criterion for the license check of Flexlm (AND / OR)</p> <p>AND: Recognized as "license available" only when ALL features specified by both LM_FROM_FEATURE and LM_TO_FEATURE are available</p> <p>OR: Recognized as "license available" when at least one features specified by LM_FROM_FEATURE or LM_TO_FEATURE is available Please note that it will be regarded as "OR" when LM_FEATURE_OPERATOR is not specified.</p> <p>Note: This parameter needs to be configured along with LM_FROM_FEATURE, LM_TO_FEATURE, LM_FROM_LICENSE, LM_TO_LICENSE, LM_FROM_TRANSLATOR and LM_TO_TRANSLATOR.</p>
Example	LM_FROM_FEATURE_OPERATOR AND

LM_LICENSE_WAIT_COUNT

Description	<p>The number of retry for the case when none of the license is available after the license check of Flexlm. Retry will be performed for the number of the times specified at "LM_LICENSE_WAIT_COUNT". Please note that it will be regarded as "0 (no retry)" when this is not set.</p>
Example	LM_LICENSE_WAIT_COUNT 10

LM_LICENSE_WAIT_INTERVAL	
Description	Specify the time interval for the license re-check on CAD licenses. (Unit: Second) License check will be repeated for the times specified with "LM_LICENSE_WAIT_COUNT" by this time interval. This is effective only when the value for "LM_LICENSE_WAIT_COUNT" is 1 or greater. It will be set to "30 seconds" when this is not set.
Example	LM_LICENSE_WAIT_INTERVAL 30
LICENSE_WAIT_COUNT	
Description	The retry counts for CAD to ENF, Optimizer, and ENF to CAD. If it is not specified, the count is zero.
Example	LICENSE_WAIT_COUNT 5
LICENSE_WAIT_INTERVAL	
Description	Specify the time interval for the license re-check on Elysium licenses. (Unit: Second) License check will be repeated for the times specified with "LICENSE_WAIT_COUNT" by this time interval. This is effective only when the value for "LM_LICENSE_WAIT_COUNT" is 1 or greater. It will be set to "30 seconds" when this is not set.
Example	LICENSE_WAIT_INTERVAL 30
LICENSE_DETAIL	
Description	Elybatch exports a detailed log file returned from Elysium License Server.
Example	LICENSE_DETAIL
BATCH_TIME_OUT	
Description	This is maximum limit of translation running time. If the translation has not finished during specified time, all related processes are stopped and killed. (Default: 360 minutes: 6 hours)
Example	BATCH_TIME_OUT 120
LEGACY_MODE	
Description	An option setting whether to use legacy adapter (0/1) 0: Use the current adapter 1: Use legacy adapter Default setting differs for each CAD and the components.

Example	LEGACY_MODE 1
NUM_OF_PROCESSES	
Description	<p>Specify the maximum number of processes available for translation. When no value is specified, an appropriate value will be applied in accordance with the corresponding adapter.</p> <p>When zero (0) is specified, an available maximum number of processes will be applied (in accordance with the environment.)</p> <p>This option is available only when it satisfies the following conditions:</p> <ul style="list-style-type: none"> - LEGACY_MODE is not specified or 0 is specified for it. - ENF Reader of NX, CATIA V5, or JT. - ENF Writer of PLM XML and STEP AP242 BOM.
Example	NUM_OF_PROCESSES 4

Each Translation Settings (Product Sensitive)

3DEX_Repository	
Description	Available for 3DEXPERIENCE ENF Writer/Reader. Specify the repository name of 3DEXPERIENCE provider used during translation. If unspecified, PLM1 is applied.
Example	3DEX_Repository PLM1
3DEX_Server	
Description	Available for 3DEXPERIENCE ENF Writer/Reader. Specify the URL of 3DSpace in the following format: <protocol>://<HostName>:<Port>/<RootURI>
Example	3DEX_Server https://servermachine.mycompany.co.jp:8080/3dspace
3DEX_LoginTicket	
Description	Available for 3DEXPERIENCE ENF Writer/Reader. Specify the login ticket string of 3DEXPERIENCE.
Example	3DEX_LoginTicket NEE3QkYxNzY4N0MyNDJDMEFEMzVFNUU5NDE4MEEyNUV8ZGV2MjAxN 3h8ZGV2 (* No line break is required.)
3DEX_PLMType	
Description	Available for 3DEXPERIENCE ENF Writer/Reader. Specify the PLM type. If unspecified, VPMReference is applied.
Example	3DEX_PLMType VPMReference

3DEX_EnvPath	
Description	Available for 3DEXPERIENCE ENF Writer/Reader. Specify the 3DEXPERIENCE env file by its absolute file path to apply a customized configuration. If unspecified, the default env file is applied.
Example	3DEX_EnvPath C:\Program Files\Dassault Systemes\B420\CATEnv\Env.txt
COLOR_DEF_FILE	
Description	Available for NX ENF Writer. It sets color definition file (*.cdf).
Example	COLOR_DEF_FILE D:\data\ugcolor_sample.cdf
WIDTH_DEF_FILE	
Description	Available for NX ENF Writer. It sets line width mapping file (*.txt)
Example	WIDTH_DEF_FILE D:\data\width_def_file.txt
PARASOLID_SCHEMA	
Description	Required for Parasolid ENF Reader/Writer. Path to the schema directory in Elysium installation.
Example	PARASOLID_SCHEMA \$ESERVER\ParasolExe\schema
OPT_FROM_TRANSLATOR	
Description	Required for Creo Parametric to ENF translation.
Example	OPT_FROM_TRANSLATOR pro_wait
OPT_TO_TRANSLATOR	
Description	Required for ENF to Creo Parametric translation.
Example	OPT_TO_TRANSLATOR pro_wait
PRO_NMSD_FINISH	
Description	Available for Creo Parametric ENF Reader/Writer. It aborts the nmsd.exe (Creo Parametric startup monitoring daemon) when Elybatch process is stopped.
Example	PRO_NMSD_FINISH
PRO_PROD_DCU_FINISH	
Description	Available for Creo Parametric ENF Reader/Writer. It aborts the prod_dcu.exe (Process related with PTC Quality Agent) when Elybatch process is stopped.

Example	PRO_PROD_DCU_FINISH 1
CATIAV5_EnvPath	
Description	Available for CATIA V5 ENF Reader/Writer. Specify the CATIA V5 env file by its absolute file path to apply a customized configuration. If unspecified, the default env file is applied.
Example	CATIAV5_EnvPath C:\ProgramData\DassaultSystemes\CATEnv\CATIA.V5-6R2018.B28_MyEnv.txt (* No line break is required.)
CAA_WAIT_COUNT	
Description	Available for CATIA V5 ENF Reader/Writer. It sets the counts of retry to launch CATIA since it takes time to startup. (Default: 30 times)
Example	CAA_WAIT_COUNT 30
CAA_WAIT_INTERVAL	
Description	Available for CATIA V5 ENF Reader/Writer. It sets the interval during the retry set by CAA_WAIT_COUNT above.
Example	CAA_WAIT_INTERVAL 10
CAA_LICENSE_WAIT_COUNT	
Description	When a license fails to be obtained, the translation of CATIA V5 ENF Reader/Writer will be retried. On CAA_LICENSE_WAIT_COUNT, you can specify the number of retry. If it is not specified, retry will not be performed.
Example	CAA_LICENSE_WAIT_COUNT 10
CAA_LICENSE_WAIT_INTERVAL	
Description	Interval between the retries specified by CAA_LICENSE_WAIT_COUNT can be changed by seconds. This setting is valid only when the number of retry is more than one. If it is not specified, 30 seconds will be set.
Example	CAA_LICENSE_WAIT_INTERVAL 30
TEMPLATE_ASSEMBLY	
Description	Available for CATIA V5, Creo Parametric and NX ENF Reader. It sets template file which is used for assembly file.
Example	TEMPLATE_ASSEMBLY C:\data\template\template.asm.1
TEMPLATE_PART	

Description	Available for CATIA V5, Creo Parametric and NX ENF Reader. It sets template file which is used for part file.
Example	TEMPLATE_PART C:\data\template\template.prt.1

2.2.2. Reference Keywords and Use Environment Variables

In the configuration file, it is possible to refer to configuration keywords and environment variables as external variables. Variables that can be referenced are as follows.

- ESERVER written in the common section of the configuration file
- Environment variables which are specified before Elybatch is called

If you write "\$Variable", the value of the variable will be referenced.

Example

- (Environment Variables)

```
CADVER=21
```

- (Common Settings in configuration file)

```
ESERVER      C:\Asfalis\bin
```

- (PRODUCT Settings in configuration file)

```
PRODUCT      117001
FROM_TRANSLATOR $ESERVER\CaaExe\win\B$CADVER
```

- When FROM_TRANSLATOR is specified as above, the setting of FROM_TRANSLATOR has the same meaning as below.

```
PRODUCT      117001
FROM_TRANSLATOR C:\Asfalis\bin\CaaExe\win\B21
```

2.2.3. Multiple Entries in Configuration File

The argument "-T" enables to use different settings for a single Product ID. For example, you can include multiples CAD version settings in a single file.

- Example)

PRODUCT	117001
FROM_TRANSLATOR	C:\Asfalis\bin\CaaExe\win\B21
PATH_FROM_TRANSLATOR (snip)	C:\Program Files\Dassault Systemes\B21\intel_a\code\bin
PRODUCT	117001 22
FROM_TRANSLATOR	C:\Asfalis\bin\CaaExe\win\B22
PATH_FROM_TRANSLATOR (snip)	C:\Program Files\Dassault Systemes\B22\intel_a\code\bin
PRODUCT	117001 20
FROM_TRANSLATOR	C:\Asfalis\bin\CaaExe\win\B20
PATH_FROM_TRANSLATOR (snip)	C:\Program Files\Dassault Systemes\B20\intel_a\code\bin

In case that the argument "-T" is not specified, translation will be executed using CATIA V5R21.

In case that following command line is specified, translation will be executed using CATIA V5R20.

```
elybatch.exe *** -p117001 -T20 ***
```

2.2.4. Samples

- Case 1) CATIA V4 to NX I-deas translation

ESERVER	C:\Asfalis
PORT_NUM	5093
LIC_SERVER	XXX
TESSELLATOR	\$ESERVER\tools\win
PRODUCT	106031
FROM_TRANSLATOR	\$ESERVER\Cat4Exe\win
ADEK_DIR	\$ESERVER\tools\win
TO_TRANSLATOR	\$ESERVER\IdeasExe\win\id14
TO_CAD_START_CMD	C:\NXi6\bin\I-DEASOGL.cmd
BATCH_STATUS	

- Case 2) Creo Parametric to IGES translation

```

ESERVER          C:\Asfalis
PORT_NUM         5093
LIC_SERVER        XXX
TESSELLATOR      $ESERVER\tools\win

PRODUCT          104991
FROM_TRANSLATOR   $ESERVER\ProeExe\win\creo40
FROM_CAD_START_CMD C:\Program Files\PTC\Creo 4.0\F000\Parametric\bin
\parametric.bat
ADEK_DIR          $ESERVER\tools\win
TO_TRANSLATOR     $ESERVER\igesExe\win
BATCH_STATUS

```

2.3. Command Line Arguments

2.3.1. Command Line Arguments

See the Elybatch program sample usage below.

```

elybatch.exe -f<Configuration file> -p<Product ID> -w<Working Directory> -i<Input
File> -o<Output File> -u<Log File> -c<Parameter File> -n<PDQ Checker>

```

-f	[Mandatory] Configuration file. (Absolute file path)
-w	[Mandatory] Working directory. (Absolute file path. UNC path is not supported.)
-p	[Mandatory when the XML scenario is not used] Product ID Do not give argument "-p" when an optimizer is executed.
-i	[Mandatory when the XML scenario is not used] Input file name. Input file name of source CAD system or ENF (*.enf) file name. (*1) File must be located in working directory specified by -w argument.
-o	[Mandatory when the XML scenario is not used] Output file name. Output file name of destination CAD system (*1) Use ENF (*.enf) file name if you use the Stop process option. (-s, -b, -a option below) Output file is exported in working directory specified by -w argument.

-u	Log file name. (Absolute file path or relative file path from working directory.) In case that the filename specified by argument "-u" differs from the value in <logfile> tags in XML scenario, the logs will be exported to the file specified by <logfile> tags during the process of components in the scenario.
-X	Export log file in XML format.
-c	Parameter file (Absolute file path)
-s	Stop option: Used with either of -b or -a option below.
-b	Before Healing stop option If you specify -s -b argument, translation process stops before healing process and export ENF. Note: Need to input ENF name in -o argument.
-a	After Healing stop option If you specify -s -a argument, translation process stops after healing process and export ENF. Note: Need to input ENF name in -o argument.
-k	Skip Healing option It skips healing process. Specify ENF name by -i argument.
-n	PDQ Checker option Need to add NUT_CHECK_DIR in configuration file. (*2)
-t	Timeout option (per minute) Default: 360 min (6hours) This option has higher priority than setting in configuration file. (BATCH_TIME_OUT)
-T	Multiple Configurations option It enables to define multiple configurations by this option. (1 or larger integral number to be set) See Configuration File Variation.
-S	XML scenario file (Absolute file path)
-U	Scenario log file (Absolute file path)

- (*1) Depending on ENF Writer/Reader, the behavior is different. See "Elybatch Component Adapter" for details.
- (*2) Depending on Stop option and healing option, the behavior is different. See "Elybatch Component PDQ Checker" for details.

2.3.2. Sample Command Line Arguments

1.

Convert CATIA V4 file (test.model) to Parasolid file (test.x_t).	
elybatch.exe -felybatch.cfg -p106051 -itest.model -otest.x_t -wC:\elybatch\work	
2.	Convert Creo Parametric file (test.prt) to Parasolid file (test.x_t) along with a log file (test.log).
	elybatch.exe -felybatch.cfg -p104051 -itest.prt -otest.x_t -utest.log -wC:\elybatch\work
3.	Convert CATIA V4 file (test.model) to ENF flavored for NX I-deas without healing process.
	elybatch.exe -felybatch.cfg -p106031 -itest.model -otest.enf -s -b -wC:\elybatch\work
4.	Convert CATIA V4 file (test.model) to ENF flavored for NX I-deas after healing process.
	elybatch.exe -felybatch.cfg -p106031 -itest.model -otest_cl.enf -s -a -wC:\elybatch\work
5.	Convert ENF flavored for NX I-deas (test.enf) that is not healed to CATIA V4 file (test.model)
	elybatch.exe -felybatch.cfg -p103061 -itest.enf -otest.model -wC:\elybatch\work
6.	Convert the unhealed ENF out of NX I-deas (test.enf) to CATIA V4 file (test_cl.enf) after healing process.
	elybatch.exe -felybatch.cfg -p103061 -itest.enf -otest_cl.enf -s -a -wC:\elybatch\work

2.4. XML Scenario

Elybatch calls the components based on the execution orders defined in an external file. The external file needs to be described in the XML format.

Following components can be invoked by "Scenario Control."

CAD2ENF	Convert CAD file to neutral file. (ENF Writer)
ENF2CAD	Convert neutral file to CAD file. (ENF Reader)
DEK	Run healing process.
PDQ	Run PDQ check.
SIMP	Run Geometry Simplification.
GEOMDIFF	Run CAD Comparison.
ATTREDIT	Update attribute in neutral file.
ASMEDIT	Edit Assembly Structure.
INTRCHK	Check interference between parts.
POLYGON	Create polygon entity.

PLYGNOPTIMIZE	Run Polygon Optimizer.
ENFEDITOR	Edit the information on ENF file.
DSCHECK	Run DFM Studio Checker.
DSREPORT	Run DFM Studio Reporter.

2.4.1. Format for the Scenarios

I. Format for the Scenarios

Character code is not specified, but you should declare the character code for the file according to XML format.

Relative path for the desired folder should be specified relatively from the working folder.

- XML file structure
(See following sections for details on each component.)
 - 1 . Header
 - 2 . ScenarioList
 - 3 . Scenario
 - 4 . Tag (for each component)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--RootElement -->
<ScenarioList>
  <!--Scenario Element -->
  <Scenario>
    <!-- Components -->
    <CAD2ENF>
    </CAD2ENF>
    <DEK>
    </DEK>
  </Scenario>
</ScenarioList>
```

- External Reference
It is possible to set arguments and environment variables by external variables.

```
${ Variable KEY } = Variable VALUE
```

The KEY needs to be defined by capitals in \${ }.

- Reserved Words
Following values are defined as reserved words.
Environment variable has the higher priority when the value in an environment variable

and the reserved word is duplicated.

<code>\${INPUTFILE}</code>	File name set by <code>-i</code> argument (Incl. file path)
<code>\${INPUTPATH}</code>	Path set by <code>-i</code> argument
<code>\${INPUTNAME}</code>	File name set by <code>-i</code> argument
<code>\${INPUTTEXT}</code>	File extension set by <code>-i</code> argument (exclude period)
<code>\${INPUTFILE} = \${INPUTPATH}\\${INPUTNAME}.\${INPUTTEXT}</code>	
<code>\${OUTPUTFILE}</code>	File name set by <code>-o</code> argument (Incl. file path)
<code>\${OUTPUTPATH}</code>	Path set by <code>-o</code> argument
<code>\${OUTPUTNAME}</code>	File name set by <code>-o</code> argument
<code>\${OUTPUTTEXT}</code>	File extension set by <code>-o</code> argument (exclude period)
<code>\${LOGFILE}</code>	File name set by <code>-u</code> argument (Incl. file path)
<code>\${PARAMETERFILE}</code>	File name set by <code>-c</code> argument (Incl. file path)
<code>\${WORKDIR}</code>	Working directory set by <code>-w</code> argument
<code>\${CONFIGFILE}</code>	File name set by <code>-f</code> argument (Incl. file path)
<code>\${XMLLOGFILE}</code>	File name set by <code>-x</code> argument (Incl. file path)
<code>\${PRODUCTID}</code>	Product ID specified by <code>-p</code> argument
<code>\${PREPRODUCTID}</code>	Product ID of Source CAD specified by <code>-p</code> argument
<code>\${POSTPRODUCTID}</code>	Product ID of Target CAD specified by <code>-p</code> argument
<code>\${TIMEOUT}</code>	Timeout set by <code>-t</code> argument

- Compatibility with arguments

When you specified Scenario XML file, arguments are used as external reference and former arguments cannot be used.

Also the arguments, `-s`, `-b`, `-a`, `-k`, `-n` are disregarded when you use Scenario XML.

II. Scenario Tag

- Scenario Tag

Section: CAD2ENF

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M
<productcode>	Product	M	id	Specify ID. It is always "xxx001" for ENF Writer	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<parameterfile>	Parameter File	O	path	Absolute /Relative path	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: ENF2CAD

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M
<productcode>	Product	M	id	Specify ID. It is always "xxx001" for ENF Writer	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<parameterfile>	Parameter File	O	path	Absolute /Relative path	M

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: DEK

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M
<productcode>	Product	M	id	Specify ID. It is always "xxx001" for ENF Writer	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<parameterfile>	Parameter File	O	path	Absolute /Relative path	M
			productcode	Use the corresponding parameter to "productcode" tag	O
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: PDQ

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<csvfile>	Output file in CSV	M	path	Absolute /Relative path	M
<outputfile>	Export ENF file that includes check result	O	path	Absolute /Relative path	M
<formatfile>	Format File	M	path	Absolute /Relative path	M
<configfile>	Config file	M	path	Absolute /Relative path	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<parameterfile>	Parameter file for check categories	M	path	Absolute /Relative path	M
<logfile>	Log file in text format	O	path	Exports in stdout if it is not specified	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: SIMP

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<parameterfile>	Parameter File	M	path	Absolute /Relative path	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: GEOMDIFF

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<sourcefile>	Source file of comparison	M	path	Absolute /Relative path	M
<targetfile>	Target file of comparison	M	path	Absolute /Relative path	M
<diffresultfile>	verification result file	O *1	path	Absolute /Relative path	M
<parameterfile>	Parameter	O	path	Absolute /Relative path	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O *1	path	Absolute /Relative path	M
<workdir>	Work directory	M	path	Absolute path UNC path is not supported	M
<timeout>	Timeout	O	value	Value is in Minute	M
<xmlreportfile>	Validation report file (XML) (*2) (*5)	O	path	Absolute /Relative path	M
<html3dreportfile>	Validation report file (3D HTML) (*3) (*5)	O	path	Absolute /Relative path	M
<pdf3dreportfile>	Validation report file (3D PDF) (*4)	O	path	Absolute /Relative path	M
<diffanalyzer>	Validation report file (Self-contained visualization file / exe file) (*6)	O	path	Absolute /Relative path	M

- (*1) Either one is to be defined.
- (*2) Available only when CreateReport=1 is specified.

If the parameter CreateReport=1 is specified and the tag <xmlreportfile> is not specified, a report file name is automatically determined based on the name of the <diffresultfile> and <xmllogfile>.

Please note that the report will not be exported when there is a file or a folder under the same file path (without file extension) as what you specified with "xmlreportfile".

- (*3) Available only when "Create3DReport=1" is specified.

If the parameter "Create3DReport=1" is specified and the tag <html3dreportfile> is not specified, a report file name will be automatically determined based on the name of the <diffresultfile> and <xmllogfile>.

Please note that the report will not be exported when there is a file or a folder under the same file path (without file extension) as what you specified with "html3dreportfile".

- (*4) Available only when "Create3DPdfReport=1" is specified.

If the parameter "Create3DPdfReport=1" is specified and the tag <pdf3dreportfile> is not specified, a report file name will be automatically determined based on the name of the <diffresultfile> and <xmllogfile>.

Please note that the report will not be exported when there is a file under the same file path as what you specified with "pdf3dreportfile".

- (*5) Please ensure to specify a different folder as the output folder for XML and 3D HTML validation reports. (You cannot specify the same file path with different file extensions to parameter "xmlreportfile" and "html3dreportfile".) This is because a subfolder will be created for each format (XML and 3D HTML) in the output folder under the same name as the report file.)
- (*6) Available only when "CreateDiffAnalyzer=1" is specified.

If the parameter "CreateDiffAnalyzer=1" is specified and the tag <diffanalyzer> is not specified, a report file name will be automatically determined based on the name of the <diffresultfile> and <xmllogfile>.

Please note that the report will not be exported when there is a file under the same file path as what you specified with "diffanalyzer".

Section: ATTREDIT

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<bomfile>	Output File of attribute	O	path	Absolute/Relative path	M
<maprfile>	Mapping File	M	path	Absolute /Relative path	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: ASMEDIT

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M
<parameterfile>	Parameter	M	path	Absolute /Relative path	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<namelist>	Specify delete assembly/part list	O *1	path	Absolute/Relative path	M
<extraction_target>	Specify XML file for extraction assembly/part list	O *2	path	Absolute/Relative path	M
<extraction_outputdir>	Specify output folder for extraction	O *2	path	Absolute/Relative path	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

- (*1) mandatory for delete part/assembly.
- (*2) mandatory for extract part/assembly.

Section: INTERCHK

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<csvfile>	Output CSV File	M	path	Absolute /Relative path	M
<parameterfile>	Parameter	M	path	Absolute /Relative path	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: POLYGON

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M
<parameterfile>	Parameter File	M	path	Absolute /Relative path	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: PLYGNOPTIMIZE

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<inputfile>	Input File	M	path	Absolute /Relative path	M
<outputfile>	Output File	M	path	Absolute /Relative path	M

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M
<parameterfile>	Parameter File	M	path	Absolute /Relative path	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

Section: ENFEDITOR

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<scriptfile>	Ruby script file for enf editor	M	path	Absolute path	M
<inputfile>	Input file	O *1	path	Absolute /Relative path	M
<outputfile>	Output file	O *2	path	Absolute /Relative path	M
<bomfile>	Output file of attribute	O *3	path	Absolute /Relative path	M
<maprfile>	Attribute mapping file	O *3	path	Absolute /Relative path	M
<namelist>	Specify delete assembly/part list	O *4	path	Absolute /Relative path	M
<extraction_target>	Specify XML file for extraction assembly/part list	O *4	path	Absolute /Relative path	M
<extraction_outputdir>	Specify output folder for extraction	O *4	path	Absolute /Relative path	M
<workdir>	Working directory	M	path	Absolute path UNC path is not supported	M

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<parameterfile>	Parameter file	O	path	Absolute /Relative path	M
<logfile>	Log file in text format	O	path	Absolute /Relative path	M
<xmllogfile>	Log file in XML format	O	path	Absolute /Relative path	M
<timeout>	Timeout	O	value	Value is in Minute	M

- (*1) This is mandatory when specifying a pre-installed scripts as the scriptfile. This is for both editing assemblies and editing attributes.
- (*2) This is mandatory when specifying a pre-installed scripts as the scriptfile. This is only for editing assemblies.
- (*3) This is effective only when specifying a pre-installed scripts as the scriptfile. This is only for editing attributes.
- (*4) This is effective only when specifying a pre-installed scripts as the scriptfile for editing assemblies.

Section: DSCHECK

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<scriptfile>	Ruby script file for DFM Studio Checker	M	path	Absolute path	M
<inputfile>	Input file	M	path	Absolute path	M
<outputfile>	Output file	M	path	Absolute path	M
<workdir>	Work folder	M	path	Absolute path	M
<parameterfile>	Parameter file	O	path	Absolute path	M
<logfile>	Log file in text format	O	path	Absolute path	M
<xmllogfile>	Log file in XML format	O	path	Absolute path	M
<timeout>	Timeout	O	Value	Value is in Minute	M

Section: DSREPORT

Availability (M: Mandatory / O: Option)

Tag	Description	M/O (Tag)	Attributes	Description	M/O (Attr.)
<scriptfile>	Ruby script file for DFM Studio Reporter	M	path	Absolute path	M
<inputfile>	Input file	M	path	Absolute path	M
<workdir>	Work folder	M	path	Absolute path	M
<parameterfile>	Parameter file	O	path	Absolute path	M
<logfile>	Log file in text format	O	path	Absolute path	M
<xmllogfile>	Log file in XML format	O	path	Absolute path	M
<timeout>	Timeout	O	Value	Value is in Minute	M

2.4.2. Execution Example

I. Specifying the XML scenario and scenario log file

- XML Scenario File Settings

Define the path to XML scenario file that is commonly used for the common section in the configuration file.

```
SCENARIO_FILE [File path to XML scenario file]
```

Or, it can be set by passing -S argument to Elybatch.

- Scenario Log File

It is possible to export a log file by setting the following keywords in configuration file

```
SCENARIO_LOG [File path]
```

or -U argument to Elybatch.

- Sample

```

<SCENARIO_LOG>
  <PROGRESS>
    <totalstep count=1>
    <currentstep count=1>
  </PROGRESS>
  <DEK>
    <returncode value="1002"/>
    <inputfile path="XXXX"/>
    .....
  </DEK>
</SCENARIO_LOG>

```

- Terms of repeated process

Determines the last execute return code, and if an error is detected, the scenario will be aborted at that point.

(However, processing will continue for error code 1000-1100.)

II. Execution Example

Example (a)

- Sample scenario

```

<?xml version="1.0" encoding="UTF-8"?>
  <ScenarioList>
    <Scenario>
      <CAD2ENF>
        <inputfile path="C:\in\test.CATPart"/>
        <outputfile path="C:\out\test.enf"/>
        <productcode id="117001"/>
        <parameterfile path="C:\param\param.txt"/>
        <logfile path="test.log"/>
        <xmllogfile path="test.xml"/>
        <workdir path="C:\work"/>
      </CAD2ENF>
    </Scenario>
  </ScenarioList>

```

- Sample command

```

"C:\ASFALIS\bin\tools\win\elybatch.exe" -f"C:\config\elybatch.cfg" -p117001
-w"C:\work" -S"C:\scenario\scenario.xml"

```

Example (b)

- Sample scenario

```
<?xml version="1.0" encoding="UTF-8"?>
<ScenarioList>
  <Scenario>
    <CAD2ENF>
      <inputfile path="${INPUTFILE}"/>
      <outputfile path="${INPUTNAME}.enf"/>
      <productcode id="${PREPRODUCTID}001"/>
      <parameterfile path="${PARAMETERFILE}"/>
      <logfile path="${INPUTNAME}.log"/>
      <xmllogfile path="${INPUTNAME}.xml"/>
      <workdir path="${WORKDIR}"/>
    </CAD2ENF>
    <DEK>
      <inputfile path="${INPUTNAME}.enf"/>
      <outputfile path="${INPUTNAME}_cl.enf"/>
      <productcode id="100${POSTPRODUCTID}"/>
      <parameterfile path="${PARAMETERFILE}"/>
      <logfile path="${INPUTNAME}.log"/>
      <xmllogfile path="${INPUTNAME}.xml"/>
      <workdir path="${WORKDIR}"/>
    </DEK>
    <ENF2CAD>
      <inputfile path="${INPUTNAME}_cl.enf"/>
      <outputfile path="${OUTPUTNAME}"/>
      <productcode id="100${POSTPRODUCTID}"/>
      <parameterfile path="${PARAMETERFILE}"/>
      <logfile path="${INPUTNAME}.log"/>
      <xmllogfile path="${INPUTNAME}.xml"/>
      <workdir path="${WORKDIR}"/>
    </ENF2CAD>
  </Scenario>
</ScenarioList>
```

- Sample command

```
"C:\ASFALIS\bin\tools\win\elybatch.exe" -f"C:\config\elybatch.cfg" -p117141
-i"C:\in\test.CATPart" -o"Z:\out\test" -w"C:\work" -c"C:\param\param.txt"
-S"C:\scenario\scenario.xml"
```

2.5. Parameter File Settings

You can set options that are used during translation within the parameter file.
See the parameter manual for further details.

2.6. ELYBATCH Error Code

You can get the error code from the file named "elybatch.rc" that is created in the working directory when a translation has been finished.

Please refer to a separate document "ERRCODE.pdf" for the details.

2.6.1. Note

- Forced Termination

It forced to terminate the running process if you create a file "killprocess" in the working directory specified by -w argument. The error code 999 will be returned.

- Forbidden Characters check

It aborts the translation when following characters are included in the file set by "-o" argument. The error code 33 will be returned.

(It replaces the invalid characters to _ for the files used during translation.)

- WINDOWS Reserved word

If the character string till the first period "." or all character string is following characters. (Not case sensitive)

AUX, CON, NUL, PRN, CLOCK\$, COM1 - COM9, LPT1 - LPT9

- Forbidden characters in WINDOWS file

\ / : * ? " < > |

3. ASFALIS Component

3.1. Adapter

Adapters are classified into two groups by the translation method, "CAD Plug-in" and "Standalone." CAD Plug-in uses the CAD environment for translation. Standalone doesn't use the CAD environment for translation.

Adapters are classified into two groups, also by the processing method, "New Adapter" and "Legacy Adapter." New Adapter supports multi-processing. Legacy Adapter has been the standard adapters till EX5.

This chapter shows the settings and specification for each adapter.

	Type	ENF Writer		ENF Reader	
		New Adapter	Legacy Adapter	New Adapter	Legacy Adapter
CATIA V5(CAA)	Plug-in	○	◎	◎	○
3DEXPERIENCE	Plug-in	◎	-	◎	-
CATIA V4	Standalone	-	◎	-	◎
NX I-deas	Plug-in	-	◎	-	◎
Parasolid	Standalone	-	◎	-	◎
NX	Plug-in	○	◎	◎	○
Creo Parametric	Plug-in	○	◎	◎	○
Autodesk Inventor	Plug-in	△	◎	△	◎
Creo Elements/Direct	Plug-in	-	◎	-	◎
ACIS	Standalone	-	◎	-	◎
STEP	Standalone	-	◎	-	◎
STEP AP242 BOM	Standalone	◎	-	◎	-
IGES	Standalone	-	◎	-	◎
SOLIDWORKS	Plug-in	◎	○	◎	○
JT	Standalone	◎	○	◎	○
PLM XML	Standalone	◎	-	◎	-

	Type	ENF Writer		ENF Reader	
iCAD	Standalone	-	⊙	-	⊙
CATIA V5 (Standalone)	Standalone	-	⊙	-	⊙
Creo Parametric (Standalone)	Standalone	-	⊙	n/a	n/a
NX (Standalone)	Standalone	-	⊙	n/a	n/a
CADmeister (Standalone)	Standalone	-	⊙	-	⊙
XVL	Standalone	n/a	n/a	-	⊙
STL	Standalone	n/a	n/a	-	⊙
3D PDF	Standalone	n/a	n/a	-	⊙
3DXML	Standalone	-	⊙	n/a	n/a

- ⊙: default, ○: available, △: available but not supported

Hereinafter, following directory will be referred to as <ASFALIS Components>.

<ASFALIS Component package>\bin or

<ASFALIS Components installation folder>\module

3.1.1. CATIA V5 to ENF / ENF to CATIA V5

I. Necessary Executables

- **Adapter**
64bit
<ASFALIS Components>\CADFeature\CatiaV5.x64
<ASFALIS Components>\CADFeature\Batch\117
<ASFALIS Components>\CADFeature\Batch\CatiaV5.x64
<ASFALIS Components>\CADFeature\Batch\bin.x64
<ASFALIS Components>\CADFeature\common
<ASFALIS Components>\tools
<ASFALIS Components>\Cat4Exe
- **Legacy Adapter**
<ASFALIS Components>\CaaExe
<ASFALIS Components>\tools
<ASFALIS Components>\Cat4Exe

II. Configuration File Settings

- **CATIA V5 to ENF**

- **PATH_FROM_TRANSLATOR**
Installation folder of CATIA V5 (CNEXT.exe etc.)
Ex) C:\Program Files\Dassault Systemes\B24\win_b64\code\bin
- **FROM_TRANSLATOR**
Installation folder of CATIA V5 legacy adapter
Ex) C:\elysium\bin\CaaExe\win\B24
- **ENF to CATIA V5**
 - **PATH_TO_TRANSLATOR**
Installation folder of CATIA V5 (CNEXT.exe etc.)
Ex) C:\Program Files\Dassault Systemes\B24\win_b64\code\bin
 - **TO_TRANSLATOR**
Installation folder of CATIA V5 legacy adapter.
Ex) C:\elysium\bin\CaaExe\win\B24

III. Specifications of Input and Output Files

- **CATIA V5 to ENF**

In case that a model is a part, specify a CATPart file name as an input file.
In case that a model is an assembly, specify a CATProduct file name of top assembly as an input file.
Child parts and sub-assemblies should be opened normally when you open CATProduct that is imported to CATIA V5 used for translation. You can change the setting in CATIA V5 as follows:
[Tools] > [Options] > [General] > [Document] "Linked Document Localization"
- **ENF to CATIA V5**

Specify a file name without extension.
If a translated model is a part, "<specified name>.CATPart" will be generated.
If a translated model is an assembly, "<specified name>.CATProduct" and child files will be generated.

IV. Note

1. About CATIA V5, it is necessary to set the license to each login user by DSLS(Dassault Systemes License Server.) The login user who executes the adapter needs to be able to run CATIA V5.
2. The version of CATIA V5 specified by PATH_FROM_TRANSLATOR or PATH_TO_TRANSLATOR and the version of CATIA V5 Adapter specified by FROM_TRANSLATOR or TO_TRANSLATOR should be the same.
3. Please try specifying the normally-used CATIA V5 environment file as the value of keyword "CATIAV5_EnvPath" manually to the CATIA V5 adapter configuration file in case you fail to run the data translation although the installation setup is completed correctly as instructed at "II Configuration File Settings" above.
 - Ex)

Environment file name: CATIA.V5R21.B21.txt

Directory of the environment file: C:\elysium\CATEnv

Configuration file:

CATIAV5_EnvPath C:\elysium\CATEnv\CATIA.V5R21.B21.txt

Please also refer to [2.2.1, “Keyword Definitions”](#) for the details

4. To use CATIA V5 ENF Writer/Reader, allow the following programs to communicate through Windows Firewall. Change settings from Control Panel or execute the script (firewall_add.bat) as an administrator.

Adapter

CATIA V5 ENF Writer: caa2fbt.exe

CATIA V5 ENF Reader: fbt2caa.exe

The script file exists in the following path:

64bit

<ASFALIS Components>\CADFeature\Batch\CatiaV5.x64\common\firewall_add.bat

Legacy Adapter

CATIA V5 ENF Writer: caa2enf.exe

CATIA V5 ENF Reader: enf2caa.exe

The script file exists in the same folder as the modules.

V. Restriction

1. The file name of CATIA V5 cannot contain multi-byte characters and the following characters:

\ / : * ? " < > |

The following characters are allowed to use.

! # \$ % & ' () = ~ ` { + } ^ @ [] ,

- **ENF to CATIA V5**

The prohibited characters will be replaced with underscores.

Please note that, when you are translating with the legacy adapter, prohibited characters in the output filename will remain as they are, and the translation will abend when the output filename contains the prohibited characters. (The new adapter prevents this error by replacing prohibited characters with underscores.)

2. The name of part, assembly and instance of CATIA V5 cannot contain the following characters:

! :

- **ENF to CATIA V5**

The prohibited characters will be replaced with underscore.

3.1.2. 3DEXPERIENCE to ENF / ENF to 3DEXPERIENCE

I. Necessary Executables

- **Adapter**
 - <ASFALIS Components>\CADFeature\Batch\195
 - <ASFALIS Components>\CADFeature\Batch\3dex.x64
 - <ASFALIS Components>\CADFeature\Batch\bin.x64
 - <ASFALIS Components>\CADFeature\common
 - <ASFALIS Components>\tools

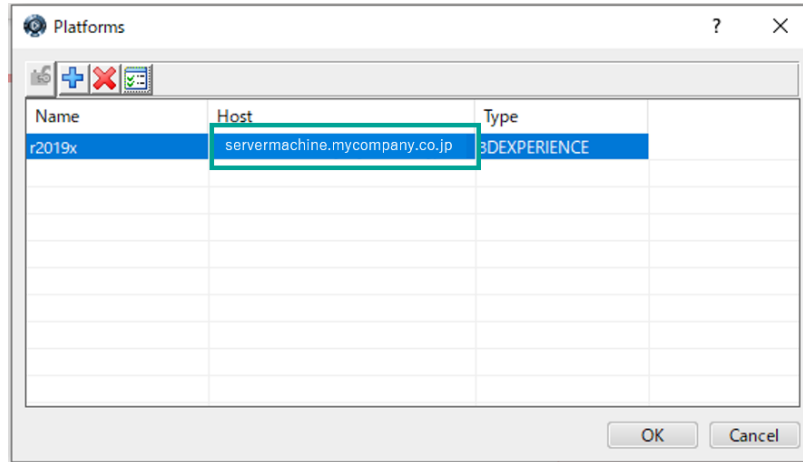
II. Configuration File Settings

- 3DEXPERIENCE to ENF
 - FROM_TRANSLATOR
CAD version name of 3DEXPERIENCE
Ex) R2018x
 - FROM_CAD_INSTALL_DIR
Installation folder of 3DEXPERIENCE Adapter
Ex) C:\Program Files\Dassault Systemes\B420\win_b64
- ENF to 3DEXPERIENCE
 - TO_TRANSLATOR
CAD version name of 3DEXPERIENCE
Ex) R2018x
 - TO_CAD_INSTALL_DIR
Installation folder of 3DEXPERIENCE Adapter
Ex) C:\Program Files\Dassault Systemes\B420\win_b64
- Common
 - 3DEX_Server
URL of 3DSpace
Ex) <https://servermachine.mycompany.co.jp:8080/3dspace>
 - 3DEX_LoginTicket
Login ticket string
Ex) NEE3QkYxNzY4N0MyNDJDMEFEMzVFNUU5NDE4MEEyNUV8ZGV2MjAxN3h8ZGV2

Please see below for how to find this information.

The URL of 3DSpace can be confirmed by either of the following ways:

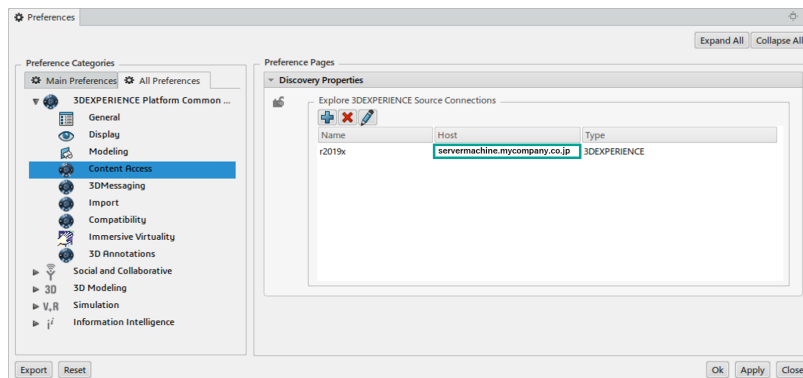
- a. Confirm in the displayed dialog when starting 3DEXPERIENCE



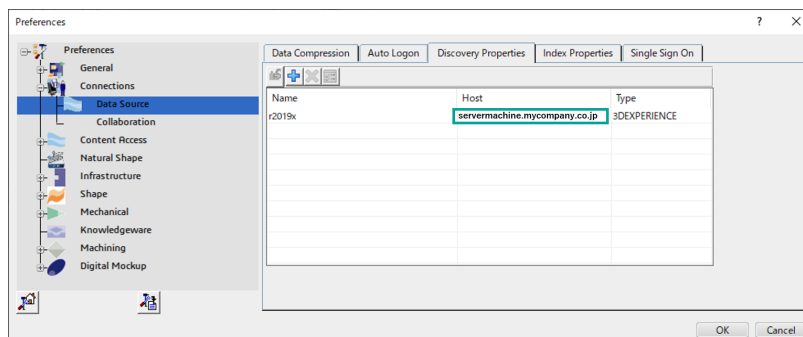
- b. Confirm in the setting window of 3DEXPERIENCE

From 3DEXPERIENCE menu, select [Preference] and you can confirm in the displayed setting window.

R2019x or later



R2018x or earlier





Login ticket string can be obtained by the following procedure:

1. Start 3DEXPERIENCE.
2. From Compass, select "My Social and Collaborative Apps".
3. Select "Collaboration & Approvals".
4. In the displayed page, select "Experience Configuration" > "Manage Login Ticket".
5. Select [Create].
6. On the displayed view, confirm the value of "Generated ticket".

Please refer to 3DEXPERIENCE manual for more details.

<https://help.3ds.com/2020x/english/DSDoc/bdvinstallmap/bdv-install-t-wsbatch.htm>

III. Specifications of Input and Output Files

• 3DEXPERIENCE to ENF

Specify a text file that describes the search condition of 3DEXPERIENCE model as an input file with parameter "-i" or <inputfile> in the Scenario. You can change the text file as follows:

- #3DEXPERIENCE (Fixed keyword, optional)
- Section Name[Model]
- Search attribute key, value

The following attribute key is available.

- PLM_ExternalID
- V_Name
- majorrevision
- minorrevision
- V_description
- owner

Example:

```
#3DEXPERIENCE
[Model]
PLM_ExternalID=prd-12345678-00000001
majorrevision=A
```

• ENF to 3DEXPERIENCE

Specify a text file as an output file with parameter "-o" or <outputfile> in the Scenario.

This text file describes the search condition of 3DEXPERIENCE model generated by translation.

IV. Restriction

1. The name of instance cannot contain the following characters:

! : \ /

- **ENF to 3DEXPERIENCE**

The prohibited characters will be replaced with underscore.

3.1.3. CATIA V4 to ENF / ENF to CATIA V4

I. Necessary Executables

- **Adapter**

<ASFALIS Components>\Cat4Exe

<ASFALIS Components>\tools

II. Configuration File Settings

- **CATIA V4 to ENF**

- FROM_TRANSLATOR

Installation folder of CATIA V4 Adapter

Ex) C:\elysium\bin\Cat4Exe\win

- **ENF to CATIA V4**

- TO_TRANSLATOR

Installation folder of CATIA V4 Adapter

Ex) C:\elysium\bin\Cat4Exe\win

III. Specifications of Input and Output Files

- **CATIA V4 to ENF**

Specify a model file name "xxx.model" as an input file.

- **ENF to CATIA V4**

Specify a model file name "xxx.model" as an output file.

IV. Note

1. How to translate a CATIA V4 model file that contains multiple models:
 - i. Get a title list file from CATIA V4 data by using "getmdl.exe."
Path: <components folder>\bin\Cat4Exe\win\getmdl.exe
Command: getmdl.exe -i<CATIA V4 data> -o<title list file>
 - ii. All model titles are written in the output title list file. Create a file with the name of "getmdl.lst" and write in the file only one title which you want to translate.
 - iii. Put "getmdl.lst" at the WORK directory and translate the CATIA V4 data. The model that

specified in getmdl.lst will be translated.

V. Restriction

1. The name of part and assembly of CATIA V4 model cannot contain multi-byte characters.
 - **ENF to CATIA V4**
The multi-byte characters will be replaced with underscore.
2. The name of part and assembly of CATIA V4 model cannot contain lowercase characters.
 - **ENF to CATIA V4**
The lowercase characters will be replaced with uppercase.
3. The number of characters of part and assembly of CATIA V4 model should be 16 or less.
 - **ENF to CATIA V4**
The characters after the 17th will be cut.

3.1.4. NX I-deas to ENF / ENF to NX I-deas

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\IdeasExe
<ASFALIS Components>\tools

II. Configuration File Settings

- **NX I-deas to ENF**
 - FROM_CAD_START_CMD
Path of NX I-deas start file.
Ex) C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
 - FROM_TRANSLATOR
Installation folder of NX I-deas Adapter
Ex) C:\elysium\bin\IdeasExe\win\id14
- **ENF to NX I-deas**
 - TO_CAD_START_CMD
Path of NX I-deas start file.
Ex) C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
 - TO_TRANSLATOR
Installation folder of NX I-deas Adapter
Ex) C:\elysium\bin\IdeasExe\win\id14

III. Specifications of Input and Output Files

- **NX I-deas to ENF**
Specify an mf1 or arc file name as an input file.

In case that you specify an mf1 file as an input file, an mf2 file should exist in the same directory as an mf1 file.

- **ENF to NX I-deas**

Specify an mf1 or arc file name as an output file.

IV. Note

1. Orbix should be running when you use NX I-deas Adapter.
2. In case that NX I-deas environment is customized, NX I-deas adapter may not be running because current directory may change in the NX I-deas start file (e.g., IDEASOGL.cmd). Please use the NX I-deas start file that is not customized after the installation is completed.
3. In case that upper limit of NX I-deas memory setting is set low, translation may not be completed. You can change memory setting in the NX I-deas parameter file that is specified by environment variable "IDEAS_PARAM<NX I-deas version>."
4. In case that the NX I-deas model file that saved by older version of NX I-deas is translated, the translation may not be completed. In such cases, save NX I-deas file again by the same NX I-deas version as adapter before translation.
5. When you use the adapter for NX I-deas 6 or later version.
When you install NX I-deas, the environment variable that is necessary to run the NX I-deas adapter has not been set. Before using the adapter, modify the value of the IDEAS_PARAM** environment variable to point to the parameter files location. For details, ask the system manager of NX I-deas.

NX I-deas 6.3	IDEAS_PARAM17=<path of the parameter file location>
NX I-deas 6.4	IDEAS_PARAM18=<path of the parameter file location>
NX I-deas 6.5	IDEAS_PARAM18=<path of the parameter file location>
NX I-deas 6.6	IDEAS_PARAM18=<path of the parameter file location>
NX I-deas 6.7	IDEAS_PARAM18=<path of the parameter file location>

V. Restriction

1. The file path to NX I-deas model files should be 80 characters at maximum.
2. The folder path to elybatch working directory should be 65 characters at maximum.
3. The file name of NX I-deas cannot contain multi-byte characters, space and the following characters:
 \ / < > | ` \$ % ^ ; : " ' * ? . &
 - **ENF to NX I-deas**
If output file name contains the prohibited characters, translation will be error.
4. The name of part and assembly of NX I-deas cannot exceed 80 characters.

- **ENF to NX I-deas**

The characters after the 80th will be cut.

5. The Bin name cannot exceed 80 characters. When only numeric characters are used for the Bin name, the name should be within 10 characters.

- **ENF to NX I-deas**

Translation will be error.

3.1.5. Creo Parametric to ENF / ENF to Creo Parametric

I. Necessary Executables

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\Batch\104

<ASFALIS Components>\CADFeature\Batch\Creo.x64

<ASFALIS Components>\CADFeature\Batch\bin.x64

<ASFALIS Components>\CADFeature\common

<ASFALIS Components>\tools

- **Legacy Adapter**

<ASFALIS Components>\ProeExe

<ASFALIS Components>\tools

II. Configuration File Settings

- **Creo Parametric to ENF**

- FROM_CAD_START_CMD

Path of Creo Parametric start file

Ex) C:\Program Files\PTC\Creo 4.0\F000\Parametric\bin\parametric.bat

- FROM_TRANSLATOR

Installation folder of Creo Parametric legacy adapter

Ex) C:\elysium\bin\ProeExe\win\creo40

- OPT_FROM_TRANSLATOR

Ex) OPT_FROM_TRANSLATOR pro_wait

- **ENF to Creo Parametric**

- TO_CAD_START_CMD

Path of Creo Parametric start file

Ex) C:\Program Files\PTC\Creo 4.0\F000\Parametric\bin\parametric.bat

- TO_TRANSLATOR

Installation folder of Creo Parametric legacy adapter

Ex) C:\elysium\bin\ProeExe\win\creo40

- OPT_TO_TRANSLATOR

Ex) OPT_TO_TRANSLATOR pro_wait



When using customized Creo Parametric start file for new adapter, the following file may need to be edited.

- <ASFALIS Components>
 \CADFeature\Batch\Creo.x64\Parametric*.0\launch_proe.bat

Example)

```
call "%PROE_INSTALL_DIR%\bin\parametric.exe"  
"%PROE_INSTALL_DIR%\bin\customized.psf" %1 %2 %3 %4 %5 %6
```

III. Specifications of Input and Output Files

• Creo Parametric to ENF

In case that a model is a part, specify a *.prt file name as an input file. In case that a model is an assembly, specify an *.asm file name of top assembly as an input file. If the extension has a numeric character such as ".1", include the number in the input file name like "*.prt.1" or "*.asm.1".

All the files (all the child parts and subassemblies) should exist in the directory where the top assembly file (the file specified as the input).

• ENF to Creo Parametric

Specify a file name without extension.

If a translated model is a part, "<specified name>.prt.1" will be generated.

If a translated model is an assembly, "<specified name>.asm.1" and child files will be generated.

IV. Note

1. We recommend using Creo Parametric that is not customized after the installation is completed, without the setting of config.pro. For example, if you set show_geom_checks_on_creation to "YES" in the config.pro, translation process may fail. If you set open_protk_signed_apps to "prompt" or "never" with Creo Parametric 7.0 or later, translation process will fail.
2. If the type of representations is Graphics Rep, translation will be error. In that case, save the model again in different type of representation.
3. Multi-processing is not supported by both legacy adapter and new adapter

V. Restriction

1. Restrictions on the filename
 - i. Filename can contain following characters only
 - In Creo Parametric 3.0 or earlier
 - Number (0 to 9)
 - Alphabet

- Hyphen "-"
- Underscore "_"

- In Creo Parametric 4.0 or later

- Number (0 to 9)
- Alphabet
- Hyphen "-"
- Underscore "_"
- Multi-byte character

Please note that the prohibited characters will be replaced by underscores "_" in ENF to Creo Parametric translation.

ii. Filename cannot start with a hyphen "-".

2. The name of part and assembly file name without extension cannot exceed 31 characters.

- **ENF to Creo Parametric**

The characters after the 31st will be cut.

If a file with the same name already exists in the work directory, branch number will be added at the end of the file name.

3. About template file specified in ENF to Creo Parametric process.

i. If "Systems of Units" contained in the template of a part file is IPS, "Systems of Units" will be not translated properly.

ii. If the "Length" of unit is different between the template file of a part and that of an assembly, layout of a translated assembly will be warped.

iii. Assembly models will be translated as follows when translated using a template file either for parts or assemblies only.

- When using a part template file only
 - Parts: Translated using a template file
 - Assemblies: Translated without a template file
- When using an assembly template file only
 - Parts: Translated without a template file
 - Assemblies: Translated using a template file

iv. Geometries such as Quilt or Datum Curve in assemblies will be translated using a part template file.

v. If user-defined length unit is used in "Systems of Units" which is specified in the template, the "Systems of Units" will be changed to "mmKs".

vi. Due to a Creo Parametric defect, "Systems of Units" may be changed to "mmKs".

4. The specification of translation of hidden solids in ENF to Creo Parametric process.

If one part has multi solids in a model, you cannot translate the no-show attribute of the solids correctly.

3.1.6. NX to ENF / ENF to NX

I. Necessary Executables

- **Adapter**
64bit
<ASFALIS Components>\CADFeature\NX.x64
<ASFALIS Components>\CADFeature\Batch\114
<ASFALIS Components>\CADFeature\Batch\NX.x64
<ASFALIS Components>\CADFeature\Batch\bin.x64
<ASFALIS Components>\CADFeature\common
<ASFALIS Components>\tools
- **Legacy Adapter**
<ASFALIS Components>\UgExe
<ASFALIS Components>\tools

II. Configuration File Settings

- **NX to ENF**
 - FROM_CAD_INSTALL_DIR
Installation folder of NX. (UGII_BASE_DIR)
Ex) C:\Program Files\Siemens\NX 10.0
 - FROM_TRANSLATOR
Installation folder of NX legacy adapter
Ex) C:\elysium\bin\UgExe\win\ug100
- **ENF to NX**
 - TO_CAD_INSTALL_DIR
Installation folder of NX. (UGII_BASE_DIR)
Ex) C:\Program Files\Siemens\NX 10.0
 - TO_TRANSLATOR
Installation folder of NX legacy adapter
Ex) C:\elysium\bin\UgExe\win\ug100

III. Specifications of Input and Output Files

- **NX to ENF**
Specify a model file name "xxx.prt" as an input file.
All the files (all the child parts and subassemblies) should exist in the directory where the top assembly file (the file specified as the input).
- **ENF to NX**
Specify a file name without extension.
If a translated model is a part, "<specified name>.prt" will be generated.
If a translated model is an assembly, "<specified name>.prt" and child files will be generated.

IV. Note

1. It is mandatory to specify the following environment variable.
To use NX9.0 or later version: SPLM_LICENSE_SERVER

V. Restriction

1. The file name of NX cannot contain multi-byte characters and the following characters when using NX9.0 or earlier version,
The file name of NX cannot contain some of the following characters when using NX10.0 or later version.
`\ / < > | ` : " * ?`
 - **ENF to NX**
The prohibited characters will be replaced with underscores.
Please note that, when you are translating with the legacy adapter, prohibited characters in the output filename will remain as they are, and the translation will abend when the output filename contains the prohibited characters. (The new adapter prevents this error by replacing prohibited characters with underscores.)
2. The name of the part and its file name without extension cannot exceed 239 characters.
 - **ENF to NX**
The characters after the maximum number will be cut.
3. Full path of the file name cannot exceed 255 characters.
 - **ENF to NX**
The characters after the maximum number will be cut.

VI. About Width Mapping File

Width mapping file is a text file for mapping line width from NX to ENF.

The file path can be specified by the configuration keyword WIDTH_DEF_FILE or parameter WidthDefFile.

- Format
<KEY>=<VALUE>
- KEY

NX_WIDTH_ORIGINAL	Corresponds to the line width of NX "No Change"
NX_WIDTH_THIN	Corresponds to the line width of NX "Thin Width"
NX_WIDTH_NORMAL	Corresponds to the line width of NX "Normal Width"
NX_WIDTH_THICK	Corresponds to the line width of NX "Thick Width"
NX_WIDTH_1_013	Corresponds to the line width of NX 1 pixels wide
NX_WIDTH_2_018	Corresponds to the line width of NX 2 pixels wide

NX_WIDTH_3_025	Corresponds to the line width of NX 3 pixels wide
NX_WIDTH_4_035	Corresponds to the line width of NX 4 pixels wide
NX_WIDTH_5_050	Corresponds to the line width of NX 5 pixels wide
NX_WIDTH_6_070	Corresponds to the line width of NX 6 pixels wide
NX_WIDTH_7_100	Corresponds to the line width of NX 7 pixels wide
NX_WIDTH_8_140	Corresponds to the line width of NX 8 pixels wide
NX_WIDTH_9_200	Corresponds to the line width of NX 9 pixels wide

- VALUE

ENF_WIDTH_0.1
ENF_WIDTH_0.2
ENF_WIDTH_0.4
ENF_WIDTH_0.6
ENF_WIDTH_0.8
ENF_WIDTH_1.0



- In case that the width mapping file is not written in the expected format defined above or the order of the key is incorrect, reading error occurs and default line widths will be used.
- Line 1 to line 13 should be written when translating.
- The lines whose width are 0.13 to 2.00mm in NX9.0 are corresponding to the lines of 1 to 9 pixels wide. Each line width of pixel is set by [Menu] > [Preferences] > [Visualization...] > [Line].

- Sample 1 (same as default setting)

```

NX_WIDTH_ORIGINAL=ENF_WIDTH_0.1
NX_WIDTH_THIN=ENF_WIDTH_0.1
NX_WIDTH_NORMAL=ENF_WIDTH_0.4
NX_WIDTH_THICK=ENF_WIDTH_1.0
NX_WIDTH_1_013=ENF_WIDTH_0.1
NX_WIDTH_2_018=ENF_WIDTH_0.4
NX_WIDTH_3_025=ENF_WIDTH_1.0
NX_WIDTH_4_035=ENF_WIDTH_1.0
NX_WIDTH_5_050=ENF_WIDTH_1.0
NX_WIDTH_6_070=ENF_WIDTH_1.0
NX_WIDTH_7_100=ENF_WIDTH_1.0
NX_WIDTH_8_140=ENF_WIDTH_1.0
NX_WIDTH_9_200=ENF_WIDTH_1.0

```

- Sample 2

```

NX_WIDTH_ORIGINAL=ENF_WIDTH_0.1
NX_WIDTH_THIN=ENF_WIDTH_0.1
NX_WIDTH_NORMAL=ENF_WIDTH_0.2
NX_WIDTH_THICK=ENF_WIDTH_0.6
NX_WIDTH_1_013=ENF_WIDTH_0.1
NX_WIDTH_2_018=ENF_WIDTH_0.2
NX_WIDTH_3_025=ENF_WIDTH_0.6
NX_WIDTH_4_035=ENF_WIDTH_0.6
NX_WIDTH_5_050=ENF_WIDTH_0.6
NX_WIDTH_6_070=ENF_WIDTH_0.6
NX_WIDTH_7_100=ENF_WIDTH_0.6
NX_WIDTH_8_140=ENF_WIDTH_0.6
NX_WIDTH_9_200=ENF_WIDTH_0.6

```

3.1.7. Parasolid to ENF / ENF to Parasolid

I. Necessary Executables

- **Adapter**
 <ASFALIS Components>\ParasolExe
 <ASFALIS Components>\tools

II. Configuration File Settings

- **Parasolid to ENF**
 - FROM_TRANSLATOR
 Installation folder of Parasolid Adapter
 Ex) C:\elysium\bin\ParasolExe\win
 - PARASOLID_SCHEMA

Path to the schema directory

- **ENF to Parasolid**

- TO_TRANSLATOR

Installation folder of Parasolid Adapter

Ex) C:\elysium\bin\ParasolExe\win

- PARASOLID_SCHEMA

Path to the schema directory

III. Specifications of Input and Output Files

- **Parasolid to ENF**

Specify a file name "xxx.x_t," "xxx.x_b," "xxx.xmt_txt" or "xxx.xmt_bin" as an input file.

- **ENF to Parasolid**

Specify a file name "xxx.x_t," "xxx.x_b," "xxx.xmt_txt" or "xxx.xmt_bin" as an output file.

IV. Restriction

1. The name of part and assembly of Parasolid cannot contain the following characters:

: ; * ? " < > | / \

- **ENF to Parasolid**

The prohibited characters will be replaced with underscore.

3.1.8. SOLIDWORKS to ENF / ENF to SOLIDWORKS

I. Necessary Executables

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\SolidWorks.x64

<ASFALIS Components>\CADFeature\Batch\109

<ASFALIS Components>\CADFeature\Batch\SolidWorks.x64

<ASFALIS Components>\CADFeature\Batch\bin.x64

<ASFALIS Components>\CADFeature\common

<ASFALIS Components>\tools

- **Legacy Adapter**

<ASFALIS Components>\SwExe

<ASFALIS Components>\tools

II. Configuration File Settings

- **SOLIDWORKS to ENF**

- FROM_TRANSLATOR

Installation folder of SOLIDWORKS legacy adapter

Ex) C:\elysium\bin\SwExe\win\2013

- **ENF to SOLIDWORKS**

- TO_TRANSLATOR

Installation folder of SOLIDWORKS legacy adapter

Ex) C:\elysium\bin\SwExe\win\2013

III. Specifications of Input and Output Files

- **SOLIDWORKS to ENF**

In case that a model is a part, specify a SLDPRT file name as an input file.

In case that a model is an assembly, specify a SLDASM file name of top assembly as an input file.

- **ENF to SOLIDWORKS**

Specify a file name without extension.

If a translated model is a part, "<specified name>.SLDPRT" will be generated.

If a translated model is an assembly, "<specified name>.SLDASM" and child files will be generated.

IV. Note

1. Multi-processing is not supported by both legacy adapter and new adapter.
2. Please note that you cannot use two or more versions of SOLIDWORKS at the same time.
Also, ASFALIS does not provide a functionality to specify the SOLIDWORKS version to use for the translation. (It automatically uses the last-used version when multiple versions of SOLIDWORKS are installed on the same computer.)
3. It is required to use a new adapter to translate PMI. You cannot translate PMI on the legacy adapter.
4. It is necessary to specify the Default Templates of SOLIDWORKS before translation.
5. When UAC (User Account Control) is turned on in Windows 8.1 and Windows 10, you should set add-in program before using SOLIDWORKS adapter.
 - i. Turn UAC off temporarily and restart the computer.
 - ii. Start SOLIDWORKS to run as administrator, select [Tools]-[Macro]-[Edit] menu and open the following macro file "load_addin.swp".
<Installation folder of SOLIDWORKS Adapter>\load_addin.swp
e.g.,) D:\bin\SwExe\win\2015\load_addin.swp
 - iii. Edit load_addin.swp and replace the value of sAddinName with full path of the original dll as below.
 - Before:
Const sAddinName As String = "sw2nut.dll"
 - After (example):
Const sAddinName As String = "D:\bin\SwExe\win\2015\sw2nut.dll"
 - iv. Run load_addin.swp.
 - v. Select [Tools]-[Add-Ins] menu of SOLIDWORKS and confirm if "Sw2nutAddin" is

displayed in the Add-ins window.

- vi. Turn UAC on and restart the computer.

3.1.9. Inventor to ENF / ENF to Inventor

I. Necessary Executables

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\Inventor.x64

<ASFALIS Components>\CADFeature\Batch\142

<ASFALIS Components>\CADFeature\Batch\Inventor.x64

<ASFALIS Components>\CADFeature\Batch\bin.x64

<ASFALIS Components>\CADFeature\common

<ASFALIS Components>\tools

- **Legacy Adapter**

<ASFALIS Components>\AiExe

<ASFALIS Components>\tools

II. Configuration File Settings

- **Inventor to ENF**

- FROM_TRANSLATOR

Installation folder of Inventor legacy adapter

Ex) C:\elysium\bin\AiExe\win\2014

- **ENF to Inventor**

- TO_TRANSLATOR

Installation folder of Inventor legacy adapter

Ex) C:\elysium\bin\AiExe\win\2014

III. Specifications of Input and Output Files

- **Inventor to ENF**

In case that a model is a part, specify an ipt file name as an input file.

In case that a model is an assembly, specify an iam file name of top assembly as an input file.

- **ENF to Inventor**

Specify a file name without extension.

If a translated model is a part, "<specified name>.ipt" will be generated.

If a translated model is an assembly, "<specified name>.iam" and child files will be generated.

IV. Note

1. In case that multiple versions of Inventor are installed on the same machine, the last used

version is used for translation.

2. To use new (EX6.0 or later) Inventor adapter, register the dll for it to the registry. Execute the following batch file (regsvr**.bat) for each version as an administrator.

The script file exists in the following path:

64bit

<ASFALIS Components>\CADFeature\Inventor.x64\regsvr<version>.bat

3.1.10. Creo Elements/Direct to ENF / ENF to Creo Elements/Direct

I. Necessary Executables

- **Adapter**
 - <ASFALIS Components>\OsdExe
 - <ASFALIS Components>\tools

II. Configuration File Settings

- **Creo Elements/Direct to ENF**
 - FROM_CAD_START_CMD
Start command of Creo Elements/Direct (SolidDesigner.exe)
Ex) C:\CAD\PTC\Creo Elements\Direct Modeling 18.0\binNT\SolidDesigner.exe
 - FROM_TRANSLATOR
Installation folder of Creo Elements/Direct Adapter
Ex) C:\elysium\bin\OsdExe\win\CM18
 - MODIFY_ENF_HEADER
Set 112001 if you want to open ENF after Creo Elements/Direct to ENF on CADdoctor.
This setting is optional.
Ex) MODIFY_ENF_HEADER 112001
- **ENF to Creo Elements/Direct**
 - TO_CAD_START_CMD
Start command of Creo Elements/Direct (SolidDesigner.exe)
Ex) C:\CAD\PTC\Creo Elements\Direct Modeling 18.0\binNT\SolidDesigner.exe
 - TO_TRANSLATOR
Installation folder of Creo Elements/Direct Adapter
Ex) C:\elysium\bin\OsdExe\win\CM18

III. Specifications of Input and Output Files

- **Creo Elements/Direct to ENF**
Specify a model file name "xxx.pkg," "xxx.sda," "xxx.sdp" or "xxx.sd" as an input file.
- **ENF to Creo Elements/Direct**
Specify a model file name "xxx.pkg" as an output file.

IV. Note

1. Set the following setting in the configuration file if you want to open ENF on CADdoctor after translating CAD data from Creo Elements/Direct to ENF.

```
MODIFY_ENF_HEADER 112001
```

V. Restriction

1. The file name of Creo Elements/Direct cannot contain the following characters:
`\ / < > | : ; , " ' ? *`
2. The name of part and assembly of Creo Elements/Direct cannot contain the following characters:
`\ / < > [] | : ; , " ' ~ ^ * + = ? ! @ # $ % & space character`
 - **ENF to Creo Elements/Direct**
The prohibited characters will be replaced with underscore.
3. The name of part and assembly of Creo Elements/Direct cannot exceed 80 characters.
 - **ENF to Creo Elements/Direct**
The characters after the 80th will be cut.

3.1.11. ACIS to ENF / ENF to ACIS

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\AcisExe
<ASFALIS Components>\tools

II. Configuration File Settings

- **ACIS to ENF**
 - FROM_TRANSLATOR
Installation folder of ACIS Adapter
Ex) C:\elysium\bin\AcisExe\win
- **ENF to ACIS**
 - TO_TRANSLATOR
Installation folder of ACIS Adapter
Ex) C:\elysium\bin\AcisExe\win

III. Specifications of Input and Output Files

- **ACIS to ENF**
Specify a file name "xxx.sat" as an input file.

- **ENF to ACIS**

Specify a file name "xxx.sat" as an output file.

3.1.12. IGES to ENF / ENF to IGES

I. Necessary Executables

- **Adapter**

<ASFALIS Components>\IgesExe

<ASFALIS Components>\tools

II. Configuration File Settings

- **IGES to ENF**

- FROM_TRANSLATOR

Installation folder of IGES Adapter

Ex) C:\elysium\bin\IgesExe\win

- **ENF to IGES**

- TO_TRANSLATOR

Installation folder of IGES Adapter

Ex) C:\elysium\bin\IgesExe\win

III. Specifications of Input and Output Files

- **IGES to ENF**

Specify a file name "xxx.igs" or "xxx.iges" as an input file.

- **ENF to IGES**

Specify a file name "xxx.igs" or "xxx.iges" as an output file.

3.1.13. STEP to ENF / ENF to STEP

I. Necessary Executables

- **Adapter**

<ASFALIS Components>\StepExe

<ASFALIS Components>\tools

II. Configuration File Settings

- **STEP to ENF**

- FROM_TRANSLATOR

Installation folder of STEP Adapter

Ex) C:\elysium\bin\StepExe\win

- **ENF to STEP**

- TO_TRANSLATOR

Installation folder of STEP Adapter
Ex) C:\elysium\bin\StepExe\win

III. Specifications of Input and Output Files

- **STEP to ENF**

Specify a file name "xxx.stp", "xxx.step" or "xxx.stpZ" as an input file.

- **ENF to STEP**

Specify a file name "xxx.stp", "xxx.step" or "xxx.stpZ" as an output file.

IV. Note

1. Due to the restriction of the format of STEP, the following elements, properties and functions are available only with the version AP203 ed2, AP214 or AP242.
 - color
 - line type / line width
 - point type
 - layer
 - show / no show
 - coordinate system
 - datum axis
 - datum plane
 - output STEP file with External Part References
2. The following elements, properties and functions are available only with the version AP242.
 - PMI
 - User-defined attributes
 - Validation properties

V. Restriction

1. The file name cannot contain the following characters:
\\ / < > | : ; , " ? *
2. The name of part and assembly cannot contain multi-byte characters.
3. The name of part and assembly cannot exceed 80 characters.
 - **ENF to STEP**
The characters after the 80th will be cut.
4. According to STEP specification, filename cannot contain non-ASCII characters.
 - **ENF to STEP**
When translating an assembly model to a STEP file with External References, part names are normally used for the filename.
The characters will be replaced to ASCII characters in case that the name of child parts includes non-ASCII characters.
Original string will be used for the output filename even in case the specified output filename includes non-ASCII characters when translating to a single STEP file.

3.1.14. STEP AP242 BOM to ENF / ENF to STEP AP242 BOM

I. Necessary Executables

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\Batch\175

<ASFALIS Components>\CADFeature\Batch\Stepbom.x64

<ASFALIS Components>\CADFeature\Batch\bin.x64

<ASFALIS Components>\CADFeature\common

<ASFALIS Components>\tools

Please note that the following folders (either a new adapter or a legacy adapter) are also mandatory to translate a model which has external reference files in JT format.

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\Batch\184

<ASFALIS Components>\CADFeature\Batch\JT.x64

- **Legacy Adapter**

<ASFALIS Components>\JtExe

Please note that the following folders (either a new adapter or a legacy adapter) are also mandatory to translate a model which has external reference files in STEP format.

- **Adapter**

<ASFALIS Components>\StepExe

II. Configuration File Settings

- **STEP AP242 BOM to ENF**

- FROM_TRANSLATOR

No additional settings are required because this is a new adapter.

- **ENF to STEP AP242 BOM**

- TO_TRANSLATOR

No additional settings are required because this is a new adapter.

III. Specifications of Input and Output Files

- **STEP AP242 BOM to ENF**

Specify a file name "XXX.stpx" or "XXX.stpxZ" as an input file with parameter "-i" or <inputfile> in the Scenario.

To translate a model which has external reference files in JT format or STEP format, place external reference files under either the folder specified in stpx or stpxZ or the same folder as stpx or stpxZ files.

- **ENF to STEP AP242 BOM**

Specify a file name "XXX.stpx" or "XXX.stpxZ" as an output file with parameter "-o" or <outputfile> in the Scenario.

JT files or STEP files will be automatically exported to the same folder as "XXX.stpx" or "XXX.stpxZ" when exporting part geometries in JT or STEP format.

IV. Note

1. Configuration of JT component and JT adapter license are also mandatory to utilize the option to translate external reference parts in JT format. The parameter setting as follow is also mandatory.

```
[DEK]
StepBomHeal=184
[ENF2STEPBOM]
ConvertPartAsCAD=184
```

2. Configuration of STEP component and STEP adapter license are also mandatory to utilize the option to translate external reference parts in STEP format. The parameter setting as follow is also mandatory.

```
[DEK]
StepBomHeal=198
[ENF2STEPBOM]
ConvertPartAsCAD=198
```

3. You can set parameters for DEK and JT/STEP for the data translation to/from STEP AP242 BOM. To set parameters, edit the script manually as the following example.

- Example)

```
stepbom['$ENF2JT_XConvertUserProperty']=1
```

Please refer to "ScriptManual_ENG.pdf" for the basic instruction on how to edit CANVAS script.

4. The STEP adapter license will be used for the data translation to/from STEP AP242 BOM..

V. Restriction

1. STEP AP242 BOM to ENF

External reference files in the format other than JT and STEP are not supported. And external reference files including both JT and STEP are not supported.

2. ENF to STEP AP242 BOM

External reference files in the format other than JT and STEP are not supported.

3. ENF to STEP AP242 BOM

When exporting part geometries in JT file, only "Full Shatter" is supported as the export format. Please refer to "ENF to JT" section for the restrictions on translating JT files.

4. ENF to STEP AP242

Assembly / part names will be renamed automatically when there is an assembly / part that conflicts by name.

3.1.15. JT to ENF / ENF to JT

I. Necessary Executables

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\Batch\184

<ASFALIS Components>\CADFeature\Batch\JT.x64

<ASFALIS Components>\CADFeature\Batch\bin.x64

<ASFALIS Components>\CADFeature\common

<ASFALIS Components>\tools

- **Legacy Adapter**

<ASFALIS Components>\JtExe

<ASFALIS Components>\tools

II. Configuration File Settings

- **JT to ENF**

- FROM_TRANSLATOR

Installation folder of JT legacy adapter

Ex) C:\elysium\bin\JtExe\win

- **ENF to JT**

- TO_TRANSLATOR

Installation folder of JT legacy adapter

Ex) C:\elysium\bin\JtExe\win

III. Specifications of Input and Output Files

- **JT to ENF**

Specify a file name "xxx.jt" as an input file.

- **ENF to JT**

Specify a file name "xxx.jt" as an output file.

IV. Note

1. Due to the restriction of the format of JT, translating PMI is available only with file version 9.4 or later. If the file version 9.3 or former is specified and XConvert*** is specified, it will be translated as version 9.4 automatically.

V. Restriction

ENF to JT

1. The name of file without extension cannot exceed 128 characters.
(The characters after the 128th will be cut.)
2. The name of part and assembly cannot exceed 128 characters.
(The characters after the 128th will be cut.)
3. Please note that following characters are prohibited in the filename by default;
 - Multi-byte characters
 - Grave accent "`"

You can change the behavior with a parameter "ReplaceNXProhibitChar" to suit your needs.

+

Parameter Value	Description
ReplaceNXProhibitChar=0	Don't treat above mentioned characters as prohibited characters
ReplaceNXProhibitChar=1	Treat above mentioned characters as prohibited characters and replace them by underscores "_" (Same as default behavior)
ReplaceNXProhibitChar=2	Treat only grave accent "`" as a prohibited character and replace it by underscores "_"



Please note that this parameter is effective only when translating using a new JT to ENF adapter. (Data translation will end with an error when translating using a legacy adapter and the input file name contains prohibited characters.)

3.1.16. PLM XML to ENF / ENF to PLM XML

I. Necessary Executables

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\Batch\194

<ASFALIS Components>\CADFeature\Batch\Plmxml.x64

<ASFALIS Components>\CADFeature\Batch\bin.x64

<ASFALIS Components>\CADFeature\common

<ASFALIS Components>\tools

Please note that the following folders (either a new adapter or a legacy adapter) are also mandatory to translate a model which has external reference files in JT format.

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\Batch\184

<ASFALIS Components>\CADFeature\Batch\JT.x64

- **Legacy Adapter**

<ASFALIS Components>\JtExe

Please note that the following folders (either a new adapter or a legacy adapter) are also mandatory to translate a model which has external reference files in NX format.

- **Adapter**

64bit

<ASFALIS Components>\CADFeature\NX.x64

<ASFALIS Components>\CADFeature\Batch\114

<ASFALIS Components>\CADFeature\Batch\NX.x64

- **Legacy Adapter**

<ASFALIS Components>\UgExe

II. Configuration File Settings

- **PLM XML to ENF**

- FROM_TRANSLATOR

No additional settings are required because this is a new adapter.

- **ENF to PLM XML**

- TO_TRANSLATOR

No additional settings are required because this is a new adapter.

III. Specifications of Input and Output Files

- **PLM XML to ENF**

Specify a file name "xxx.plmxml" as an input file with parameter "-i" or <inputfile> in the Scenario.

To translate a model which has external reference files in JT/NX format, place external reference files under either the folder specified in plmxml or the same folder as plmxml files.

- **ENF to PLM XML**

Specify a file name "xxx.plmxml" as an output file with parameter "-o" or <outputfile> in the Scenario.

JT files will be automatically exported to the same folder as "xxx.plmxml" when exporting part geometries in JT format.

IV. Note

1. Configuration of JT/NX component and JT/NX adapter license are also mandatory to utilize the option to translate external reference parts in JT/NX format.

2. You can set parameters for DEK, JT and NX for the data translation to/from PLM XML. To set parameters, edit the script manually as the following example.

- Example)

```
plmxml['$ENF2JT_XConvertUserProperty']=1
```

Please refer to "ScriptManual_ENG.pdf" for the basic instruction on how to edit CANVAS script.

V. Restriction

PLM XML to ENF

External reference files in the format other than JT/NX are not supported.

ENF to PLM XML

External reference files in the format other than JT are not supported.

ENF to PLM XML

When exporting part geometries in JT file, only "Full Shatter" is supported as the export format. Please refer to "ENF to JT" section for the restrictions on translating JT files.

3.1.17. iCAD to ENF / ENF to iCAD

I. Necessary Executables

- **Adapter**
 <ASFALIS Components>\IcadExe
 <ASFALIS Components>\tools

II. Configuration File Settings

- **iCAD to ENF**
 - FROM_TRANSLATOR
 Installation folder of iCAD adapter
 Ex) C:\elysium\bin\IcadExe\win
- **ENF to iCAD**
 - TO_TRANSLATOR
 Installation folder of iCAD adapter
 Ex) C:\elysium\bin\IcadExe\win

III. Specifications of Input and Output Files

- **iCAD to ENF**
 Specify a file name "xxx.icd" as an input file with parameter "-i" or <inputfile> in the

Scenario.

- **ENF to iCAD**

Specify a file name "xxx.icd" as an output file with parameter "-o" or <outputfile> in the Scenario.

IV. Restriction

Installation

1. You cannot install "IcadExe" using UNC paths.
2. Please ensure that the user has the write permission on the installation folder ("IcadExe" folder) when executing iCAD adapters.

iCAD to ENF

1. Please note that parts and assemblies will be unshared after the data translation.
2. Please note that semi-transparent elements will not be translated.

ENF to iCAD

1. Please note that parts and assemblies will be unshared after the data translation.
2. Please note that parts and assemblies under the same name will be renamed.
 - Ex) Coexistence of "Assy" and "Assy" will be renamed to "Assy_Asm_1" and "Assy_Asm_2" respectively.
 - Ex) Coexistence of "Body" and "Body" will be renamed to "Body_Body_1" and "Body_Body_2" respectively.
3. Following characters are prohibited in part names / assembly names.
" | \ * ; : < > ? /
Prohibited characters will be processed as follows;
ENF to iCAD : Replaced with underscores.
4. Part names / assembly names should be within 40 bytes.
Names will be processed as follows when it is exceeding 40 bytes;
ENF to iCAD : Cut off the characters beyond that length.
5. Space and following characters are prohibited in file names except for extension.
! " \$ ' * , / : ; < = > ? \ ^ ` | ~
Prohibited characters will be processed as follows;
ENF to iCAD : Replaced with underscores.
6. File names should be within 40 bytes.
Names will be processed as follows when it is exceeding 40 bytes;
ENF to iCAD : Cut off the characters beyond that length.

3.1.18. CADmeister to ENF / ENF to CADmeister (Standalone)

I. Necessary Executables

- **Adapter**

<ASFALIS Components>\CfioExe

<ASFALIS Components>\tools

<CRESTAM>\CRESTAM

CRESTAM is a directory which contains DLLs which are necessary to run CADmeister (Standalone) module.

II. Configuration File Settings

- **CADmeister to ENF (Standalone)**

- FROM_TRANSLATOR

Installation folder of CADmeister (Standalone) Adapter
Ex) C:\elysium\bin\CfioExe\win

- **ENF to CADmeister (Standalone)**

- TO_TRANSLATOR

Installation folder of CADmeister (Standalone) Adapter
Ex) C:\elysium\bin\CfioExe\win

III. Specifications of Input and Output Files

- **CADmeister to ENF (Standalone)**

Specify a model file name "xxx.cfio" as an input file.

- **ENF to CADmeister (Standalone)**

Specify a model file name "xxx.cfio" as an output file.

IV. Note

1. Please ensure to place "CRESTAM" folder under the drive where the module of CADmeister (Standalone) Adapter itself is installed. The "CRESTAM" folder is common to be used for CADmeister adapters on the computers with Japanese OS and English OS.
2. All the followings need to be placed in the same drive; the working folder, CADmeister (Standalone) Adapter and CRESTAM.
3. When you use a computer on which CADmeister is installed, place both CADmeister Adapter and CRESTAM for ASFALIS under the drive which is different from where the module of CADmeister itself is installed. (CRESTAM should be duplicated in this case.)
4. To convert the properties by using the parameter "XConvertUserProperty," you need to prepare and place following files.
 - i. Please obtain a copy of the setting files of user attribute (e.g. bhn.std, bhn.usr, etc.) from the folder below.
<Installation drive of CADmeister>\CRESTAM\KPKG\KPG\Dic
Those files contain the information customized in accordance with users' environment.

So please ensure to obtain the right files:

- For CADmeister to ENF: the setting files of the CADmeister environment that the input CFIO file was saved
 - For ENF to CADmeister: the setting files of the CADmeister environment that the output CFIO file will be opened
- ii. Please send the whole folder, "Dic" folder obtained at (1) above to Elysium Customer Support. Elysium creates XML files for attribute conversion (attr_cfio2enf.xml and attr_enf2cfio.xml) based on the contents of the setting files of user attribute you sent.
 - iii. Replace the setting files of CRESTAM for ASFALIS with the ones from CADmeister environment (obtained at (1) above).

<Installation drive of CADmeister>\CRESTAM\KPKG\KPG\Dic\bhn.std

<Installation drive of CADmeister>\CRESTAM\KPKG\KPG\Dic\bhn.usr, etc.

↓ copy to

<CRESTAM>\KPKG\KPG\Dic\bhn.std

<CRESTAM>\KPKG\KPG\Dic\bhn.usr, etc.

- iv. Place the XML files created by Elysium (at (2) above) under the same directory as CADmeister (Standalone) Adapter.

<ASFALIS Components>\CfioExe\win\attr_cfio2enf.xml

<ASFALIS Components>\CfioExe\win\attr_enf2cfio.xml

5. CADmeister (Standalone) Adapter is supported on Japanese OS and Japanese locale, or English OS.
 6. Please ensure that the color table settings match between the computer to run ENF to CADmeister translation (Computer A) and the computer to open the translated CFIO file (Computer B). If you wish to open the result CFIO file on the computer with a customized color table, please replace the color table placed below on Computer A by that on Computer B prior to the data translation.
- <Installation folder>\CRESTAM\CUSTOM\SRC\environ.color

V. Restriction

1. Object names cannot contain control codes and the characters below.
! " \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } ~
 - ENF to CADmeister (Standalone)
They will be replaced with underscores "_".
2. Object names cannot exceed 63 bytes.
 - ENF to CADmeister (Standalone)
Cut the string after the maximum number of the characters
3. Object names cannot be duplicated.
 - ENF to CADmeister (Standalone)
Add serial numbers at the end of the name

3.1.19. CATIA V5 to ENF / ENF to CATIA V5 (Standalone)

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\Dsav5Exe
<ASFALIS Components>\tools

II. Configuration File Settings

- **CATIA V5 to ENF (Standalone)**
 - FROM_TRANSLATOR
Installation folder of CATIA V5 (Standalone) Adapter
Ex) C:\elysium\bin\Dsav5Exe\win
- **ENF to CATIA V5 (Standalone)**
 - TO_TRANSLATOR
Installation folder of CATIA V5 (Standalone) Adapter.
Ex) C:\elysium\bin\Dsav5Exe\win

III. Specifications of Input and Output Files

- **CATIA V5 to ENF (Standalone)**
In case that a model is a part, specify a CATPart file name as an input file.
In case that a model is an assembly, specify a CATProduct file name of top assembly as an input file.
- **ENF to CATIA V5 (Standalone)**
Specify a file name without extension.
If a translated model is a part, "<specified name>.CATPart" will be generated.
If a translated model is an assembly, "<specified name>.CATProduct" and child files will be generated.

IV. Note

1. If multiple processes of ENF to CATIA V5 (Standalone) are executed at the same time, the translation may not be performed correctly.

V. Restriction

1. The file name of CATIA V5 cannot contain multi-byte characters and the following characters:
\\ : * ? " < > |
2. It will be replaced to the character " · " when the multibyte characters which contain 0x5C as second byte in the output filename.
3. The folder name of output file cannot contain the multibyte character which contains 0x5C as second byte.

3.1.20. Creo Parametric to ENF (Standalone)

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\DsaproExe
<ASFALIS Components>\tools

II. Configuration File Settings

- **Creo Parametric to ENF (Standalone)**
 - FROM_TRANSLATOR
Installation folder of Creo Parametric (Standalone) Adapter
Ex) C:\elysium\bin\DsaproExe\win

III. Specifications of Input and Output Files

- **Creo Parametric to ENF (Standalone)**

In case that a model is a part, specify a *.prt file name as an input file. In case that a model is an assembly, specify an *.asm file name of top assembly as an input file. If the extension has a numeric character such as ".1," include the number in the input file name like "*.prt.1" or "*.asm.1".

All the files (all the child parts and subassemblies) should exist in the directory where the top assembly file (the file specified as the input).

3.1.21. NX to ENF (Standalone)

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\DsanxExe
<ASFALIS Components>\tools

II. Configuration File Settings

- **NX to ENF (Standalone)**
 - FROM_TRANSLATOR
Installation folder of NX (Standalone) Adapter
Ex) C:\elysium\bin\DsanxExe\win

III. Specifications of Input and Output Files

- **NX to ENF (Standalone)**

Specify a model file name "xxx.prt" as an input file.

All the files (all the child parts and subassemblies) should exist in the directory where the top assembly file (the file specified as the input).

3.1.22. ENF to STL

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\StlExe
<ASFALIS Components>\tools

II. Configuration File Settings

- **ENF to STL**
 - TO_TRANSLATOR
Installation folder of STL Adapter.
Ex) C:\elysium\bin\StlExe\win

III. Specifications of Input and Output Files

- **ENF to STL**
Specify a file name "xxx.stl" as an output file.

3.1.23. ENF to XVL

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\XvlExe
<ASFALIS Components>\tools

II. Configuration File Settings

- **ENF to XVL**
 - TO_TRANSLATOR
Installation folder of XVL Adapter.
Ex) C:\elysium\bin\XvlExe\win

III. Specifications of Input and Output Files

- **ENF to XVL**
Specify a file name without extension.
The extension of an output file in XVL will be automatically adjusted based on the setting of the translation option.
If converting to P-XVL, "<specified name>.xv3" will be generated.
If converting to V-XVL or U-XVL, "<specified name>.xv2" will be generated.
If converting to multiple files, "<specified name>.xv0" and child files will be generated.

Specify an unhealed ENF as an input file.

Healing will be performed while converting from ENF to XVL.

In case that you order healing from Elybatch for XVL, the healing in adeKernel.exe will be skipped.

IV. Restriction

1. In the following cases, a group name will be changed from original one.
Name including a character which Windows does not allow to use.
Name starting with an underscore.
Name ended with "%<NUM>".

ENF to XVL

The prohibited characters will be replaced with underscore.
Underscore of the first letter will be replaced with hyphen.
"%<NUM>" of the last letter will be replaced with "#<NUM>"

3.1.24. ENF to 3D PDF

I. Necessary Executables

- **Adapter**
<ASFALIS Components>\tools

II. Configuration File Settings

- **ENF to 3D PDF**
All the modules necessary for ENF to 3D PDF translation are placed below by the standard installation, and no further configuration is required. (No need to specify "TO_TRANSLATOR" parameter, neither.)

`<ASFALIS Components>\tools`

III. Specifications of Input and Output Files

- **ENF to 3D PDF**
Specify the output filename "*.pdf" in the XML Scenario file with <outputfile> tag. Please refer to [2.4, "XML Scenario"](#) for the details about XML Scenarios.

IV. Restriction

- Please note that this Component does not support execution with command line arguments.
- Please also check [3.3.9, "3D PDF Editor"](#) when using this Component in combination with PDF Editor Component.

3.1.25. 3DXML to ENF

I. Necessary Executables

- **Adapter**
 - <ASFALIS Components>\Dsa3dxmlExe
 - <ASFALIS Components>\tools

II. Configuration File Settings

- **3DXML to ENF**
 - FROM_TRANSLATOR
 - Installation folder of 3DXML Adapter
 - Ex) C:\elysium\bin\Dsa3dxmlExe\win

III. Specifications of Input and Output Files

- **3DXML to ENF**
 - Specify a model file name "xxx.3dxml" as an input file.

3.2. Healer, etc.

3.2.1. TOGO / CADCEUS Healer

I. Necessary Executables

<ASFALIS Components>\tools

II. Configuration File Settings

- **ENF to CAD (armo file for TOGO and CADCEUS)**
 - **PRODUCT 100211**
ENF to CAD (armo for TOGO, CADCEUS): CADCEUS is 100211, TOGO is 100231
CAD (armo for TOGO, CADCEUS) to CAD: CADCEUS is 121xx1, TOGO is 123xx1
 - **ADEK_DIR** \$ESERVER\tools\win
Path of the healing program.
 - **MODIFY_ENF_HEADER 104211**
The setting for changing the header of neutral files.
For NX I-deas / CADmeister plug-in: 103211 (TOGO: 103231)
For Creo Parametric / CADmeister plug-in: 104211 (TOGO: 104231)

3.2.2. ENF Polygon

The element of the specified neutral file can be made for another neutral file as polygon elements.

I. Necessary Executables

<ASFALIS Components>\tools

II. How to use

This function can be used by defining the XML scenario.

III. Setting

You can specify the accuracy of the polygon to be generated, output element and so on. Please refer to the parameter manual for the details.

3.3. Optimizer

3.3.1. PDQChecker

I. Getting started with PDQChecker

1. How to start using Elybatch
 - a. Start using Scenario File (Recommended)
See "ELYBATCH Scenario Control" section for the usage.
 - b. Start using command line argument.
Run elybatch.exe with -n argument
2. How to start it standalone
Run PDQChecker "checknut.exe" by following arguments.

```
checknut.exe -i<ENF1> -o<Result> -e<ENF2> -c<Param> -l<Log> -f<Format>
-X<XMLLog>
```

-i	ENF to be checked
-o	CSV file that states check result
-e	ENF included check results. The result can be reviewed by ENF Viewer.
-c	Parameter file
-l	Log file
-f	Format file: <ASFALIS Components>\tools\win\DekCheck.msg
-X	XML Log file

Need to set following license settings before running the program.

```
ELY_SEC_SERVER = <Host name installed Elysium License Server>
ELY_SEC_PORT = <Port Number (Default 5093)>
```

Return value

0	Successfully completed the Job
1	Error: Insufficient argument(s)
4	Error: Invalid argument(s)
5	Error: Failed to read the parameter file

6	Error: Input ENF file is not specified correctly
7	Error: Output file path is not specified correctly
8	Error: Format file is not specified correctly
9	Error: Failed to open the format file, or the format file is invalid
10	Error: Failed to open the log file
11	Error: Failed to open the input ENF file
12	Error: Input ENF file is invalid
15	Error: Failed to open the PDQ Check result report file (*.csv)
18	Error: Failed to open the output ENF file
21	Error: Failed to save the output ENF file to the specified folder
24	Error: Failed to save the log file to the specified folder
< 0	License error (NOTE: Any negative value indicates a license related error.)

II. PDQ Check Result Report File

You can export the check result from [PDQ Checker] Component as a report in the following formats:

- PDQ check result report (CSV)

PDQ check result will be exported in CSV file in the following format.

[Header]

Version: Shows the ASFALIS version used for PDQ Check

File Name: Shows the input file name

Execute Time: Shows the start time of the PDQ Check

[Check Results]

-- FileName : <Input folder path>*.enf --: Shows the summary of the PDQ Check result of the entire model

Category ID, Category Name, Threshold, Error Num, Severity

-- Body Name : <Body name> -- : Shows the PDQ Check result of the named body
Category ID, Category Name, Threshold, Error Num, Severity, Min, Max

* Min./Max. value shows the min./max. detected value of the corresponding PDQ error

- PDQ check result report (3D PDF)

Set the following parameter to export the result in 3D PDF reports.

Create3DPdfReport=1



- It requires the license on "enf23dpdf" as well to export the result report in 3D PDF.
- 3D PDF reports will be exported with the same name as the CSV reports. (Just changing the file extension to "*.pdf".)
- Please note that 3D PDF report export is available only when running PDQ Check using XML Scenarios. It is not available when running PDQ Check from the command line argument, or by running "checknut.exe" on its own.
- Please note that the priority will be shown as "Serious" in the report when set to a user-defined level (=other than Critical / Serious / Minor) at [ENFCHECKER_PriorityMapping] section in the parameter file.
- Set the following parameter to apply a customized format template to the 3D PDF report.

"Create3DPdfReport_Type=1"

Please note that it requires the license on "pdfeditor" as well as "enf23dpdf" to use this functionality. Please contact to Elysium for details.

III. Customize PDQ Criteria Name and Severity

You can customize the PDQ criteria name and the severity level by modifying a parameter file. Otherwise, PDQ result will be exported by the default PDQ criteria name and the severity level.

- Sample

```
[ENFCHECKER_NameMapping]
G-LO-IS = Self-intersection on edge loop (Critical)
G-FA-IS = Intersection between edge loops (Minor)
[ENFCHECKER_PriorityMapping]
G-LO-IS = Critical
G-FA-IS = Minor
```

IV. Set PDQ check categories

Open "PDQCheckSetting_ENG.xls" (<ASFALIS Component>\<64bit>\util\PdqChkr)

Enter "1" in the check categories to be checked, and press Create button to create Parameter file.

You can also check thin part using CheckThinPart parameter. Please refer parameter manual.

V. Note (When running by command line)

CSV result file (*.csv) is exported in the working directory according to following conditions.

1.	No Stop option, no Healing Skip option
	[Import file name].csv : Check result before healing [Export file name].csv : Check result after healing
2.	Stop option: by "-s -b" argument
	[Import file name].csv : Check result before healing
3.	Stop option: by "-s -a" argument (Import file: CAD file)
	[Import file name].csv : Check result before healing [Export file name].enf.csv : Check result after healing
4.	Stop option: by "-s -a" argument (Import file: ENF)
	[Export file name].enf.csv : Check result <u>after</u> healing
5.	Healing skip option: When specified ENF as import file
	No result file is exported.

VI. Content of XML Log File

Following information will be exported in XML file;

- PDQ Check settings including PDQ Check criteria and threshold
- PDQ Check result

VII. Format of XML Log File

Please refer to the ASFALIS Adapter XML Log Manual ("ASFALISLOG_ENG.pdf") – chapter 4 (basic format for PDQ check result) and 5 (optional items).

VIII. Notes on Threshold Settings

Please note that PDQ Check on following errors internally uses threshold set for other error.

- Non-smooth segments
Uses the threshold set for "Large segment gap".
("Non-smooth segments" is checked on segments which the gap in between is smaller than the threshold set for "Large segment gap".)

- **Non-tangent segments**
Uses the threshold set for "Large segment gap" and "Non-smooth segments". ("Non-tangent segments" is checked on segments the gap in between is smaller than the threshold set for "Large segment gap", and also not making an angle.)
- **Non-smooth patches**
Uses the threshold set for "Large patch gap".
("Non-smooth patches" is checked on patches which the gap in between is smaller than the threshold set for "Large patch gap".)
- **Non-tangent patches**
Uses the threshold set for "Large patch gap" and "Non-smooth patches".
("Non-tangent patches" is checked on patches which the gap in between is smaller than the threshold set for "Large patch gap", and also not making an angle.)
- **Non-smooth faces**
Uses the threshold set for "Large face gap".
("Non-smooth faces" is checked on face which the gap in between is smaller than the threshold set for "Large face gap".)
- **Non-tangent faces**
Uses the threshold set for "Large face gap" and "Non-smooth faces".
("Non-tangent faces" is checked on faces which the gap in between is smaller than the threshold set for "Large face gap", and also not making an angle.)

3.3.2. Geometry Simplifier

It simplifies specified CAD data in batch. Following features are available.

Remove Short curve	Checks for edges shorter than the specified value, and updates the topology relation between edges by connecting edges which have a gap in between etc.
Remove Sliver Face	Checks for faces narrower than the specified value, and updates the topology relation between faces by connecting faces which have a gap in between etc.
Merge Face	Merges faces, and creates a single face.
Remove Round Hole	Checks for round holes, and remove all detected round holes.
Remove Generic Hole	Checks for generic holes (all holes that are not round holes), and removes all detected generic holes.
Remove Sheet Hole	Checks for holes in non-solid models, and removes all detected holes.
Remove Fillet	Checks for fillets, and removes all detected fillets.
Remove Chamfer	Checks for chamfers, and removes all detected chamfers.

Remove Step	Checks for steps, and removes all detected steps.
Extract Round Hole	Checks for round holes, and extracts all detected round holes.
Extract Generic Hole	Checks for generic holes (all holes that are not round holes), and extracts all detected generic holes.
Remove Boss And Rib	Checks for bosses and ribs, and removes all detected bosses and ribs.
Remove Rib	Checks for ribs, and removes all detected ribs.
Remove Groove	Checks for grooves, and removes all detected grooves.
Remove Logo	Checks for logos, and removes all detected logos.
Smooth Curve Array	Re-generates edges to connect them smoothly when they are not connected smoothly.
Remove Small Part	Checks for parts smaller than the specified value, and removes all detected parts.
Remove Small Volume	Checks for volumes smaller than the specified value, and removes all detected volumes.
Remove Invisible Part	Checks for volumes that are invisible from outside, and remove all detected volumes.
Remove Hole / Groove / Protrusion	Checks for holes, grooves and protrusions, and remove all detected holes, grooves and protrusions. This is like a shrink wrap ignoring protrusions, and then remove any protrusions within the specified height.
Replace with Simple Shape	Replace volumes / parts / assemblies by a simple shape (cuboid, cylinder or extruded shape).
Envelop Solid	Creates a solid envelope (a single solid model) from an assembly model (Please note that it will fail to create a single solid model when the source assembly model contains part(s) that are not connected to, or intersecting with other parts.)
Extract Visible Face	Checks on the face visibility from the specified direction(s).

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, “XML Scenario”](#).

II. Set Simplification categories

Add simplification items and process in parameter file as shown below.

See ASFALIS Parameter manual for details.

- Example)

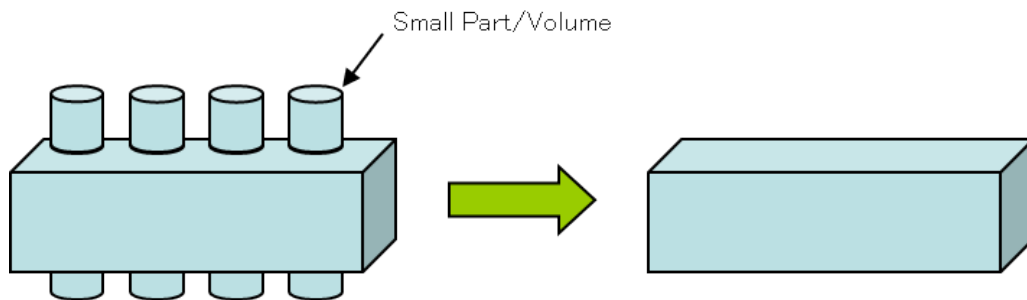
Process_0=RemoveFillet	1st step: Remove Fillet
FilletMaxRadius_0=2.5	Threshold: Fillets smaller than 2.5mm radius
Process_1=ExportENF	
Process_2=RemoveRoundHole	2nd step: Remove Round Hole
RoundHoleMaxDiameter_2=10.0	
Process_3=ExportENF	
Process_4=MergeFace	3rd step: Merge Face
MergeAngleTolerance_4=5	
ProcessNumberMax=5	Number of steps

III. About Process "Remove Small Part" / "Remove Small Volume"

Check the entire model for tiny parts and volumes that satisfies the conditions defined with parameters (e.g., the volume is smaller than the threshold), and delete the detected parts and volumes.

Please note that this may change the assembly structure because the part itself will be deleted when all the belonging volumes are deleted.

Please refer to ASFALIS Parameter manual for how to set parameters.

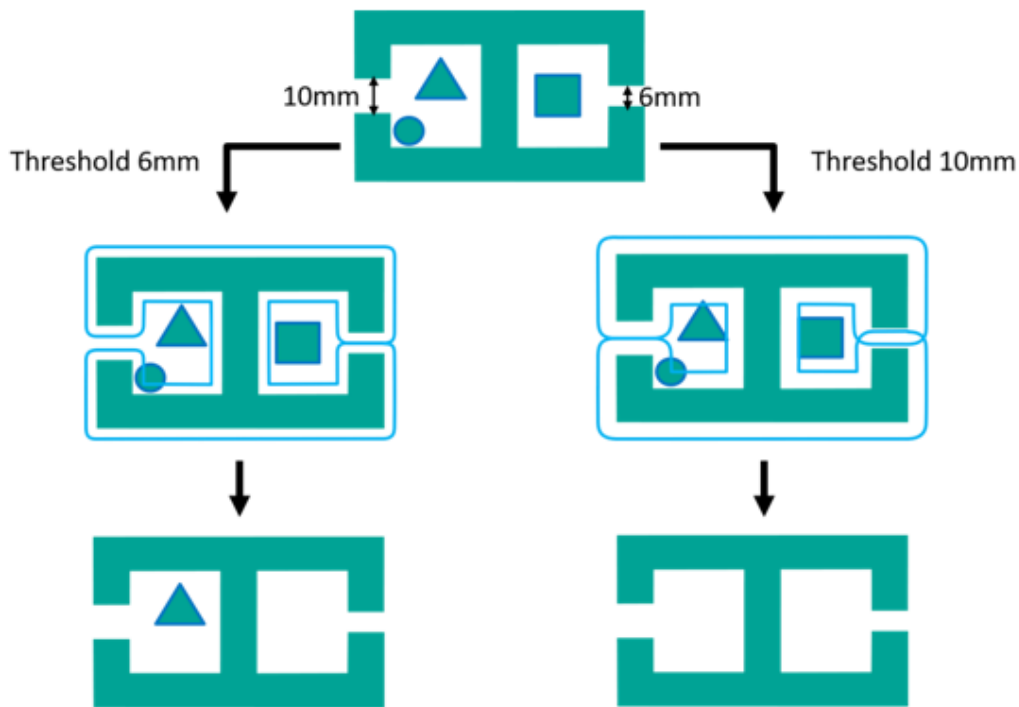


IV. Remove Invisible Part

This is a sub Component to delete all volumes for interior parts that are invisible from outside. You can also set the size of gaps to ignore so that gaps within the specified size will be treated as closed, and volumes inside that will be detected as invisible.

Please note that the part will be deleted when belonging volume(s) are all deleted, which means the tree structure will be changed.

When using [RemoveInvisiblePart] or [ExtractVisibleFace], it is recommended to run this on a computer which is equipped with graphic board that supports OpenGL version 3.0 or higher. You can still use computers without GPU; however, in that case, please note that the performance level will be the same as EX7.2.



3.3.3. CAD Validator

This is to compare two ENF files for validation / verification purposes. You can save the result in;

- "*.dvfx" files (Viewable in Model Viewer with a high viewing performance)
- "*.enf_dif" files (Viewable in CADdoctor and/or Model Viewer)
- Report files
Please refer to [VI. "Report Files"](#) for details.

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, "XML Scenario"](#).

I.I. CAD Validation Scenario

Sample Scenarios are available for CAD Validator taking two CAD files in major CAD formats as input (separately for ASFALIS Controller and ASFALIS TransServer).

- <Component folder>\sample\geomdiff
adc_scenario: Scenario for ASFALIS SmartLauncher and ASFALIS Controller
ats_scenario: Scenario for ASFALIS TransServer

I.I.I. In ASFALIS SmartLauncher

In the environment that ASFALIS SmartLauncher is installed, please place the Scenario files under the scenario folder. Please ensure to place both *.rb and *.adc files. (They function in the combination.) You can check and modify the scenario folder in ASFALIS SmartLauncher system setting.

When selecting [Run Scenario] from ASFALIS SmartLauncher(from Windows context menu), a window "Run Scenario" will appear. Specify the Scenario to run and the input model. It will be necessary to specify two models.

To create the validation report file in 3D PDF format, set the parameter "Create3DPdfReport=1" by customizing the Scenario using a text editor as follows (change the value to 1);

```
geomdiff['Create3DPdfReport'] = '1'
```

I.I.II. In ASFALIS Controller

In the environment that ASFALIS Controller is installed, please place the Scenario files under the following folder. Please ensure to place both *.rb and *.adc files. (They function in the combination.)

<ADC install folder>\data\scenario

Once set, you can select the scenario on the main dialog in ASFALIS controller. Please specify two CAD files as input.

To create the validation report file in 3D PDF format, set the parameter "Create3DPdfReport=1" by customizing the Scenario using a text editor as follows (change the value to 1);

```
geomdiff['Create3DPdfReport'] = '1'
```

Please note that Scenarios that are customized by manually editing the script files are no longer editable from the user interface of ASFALIS Controller.

I.I.III. In ASFALIS TransServer

Import the Scenario file by clicking [Import] button in "Scenario List" page to upload the Scenario file (*.json) to ASFALIS TransServer.

The scenario named "validate from <Source CAD name> to <Target CAD name>" will be created.

Please use it when executing a new job.

To create the validation report file in 3D PDF format, set the parameter "Create3DPdfReport=1".

II. Set comparison categories

Add comparison categories and process in parameter file.
See ASFALIS Parameter manual for details.

(About group flag definition file)

When grouping differences in 3D PDF report output by CAD Validator, judgment flag can be set

per group of differences. Flags selectable by default are "OK" and "Reject". Selectable flags can be customized by using the group flag definition file. In that case, users will have more options when using the report. This file should be written according to the following rules:

- Use JSON format.
- Define the array of flag object.
 - One flag is defined per one object in the array.
 - Array length must be 100 or less.
- Define the following for the flag object properties.
 - choices (mandatory)
 - Specify the option displayed in the combo box as the flag.
 - Type is Array of String. No limit for the length. (Example: ["OK", "Not Good", "Postpone"])
 - name (mandatory)
 - Specify the flag name displayed in the flag selection interface.
 - Type is String. (Example: "Approved flag")
 - approvedChoices (optional)
 - Concatenated with "choices" and displayed as options in the combo box. When the option specified in this property is selected and "Show diff only" is enabled, it will not appear in the group table.
 - Please ensure that one or less flag object in the array of flag objects has this property.
 - Type is Array of String. No limit for the length. (Example: ["Excellent", "Good", "Fair"])

When ["Excellent", "Good", "Fair"] is defined for approvedChoices, and ["Not Good", "Postpone"] is defined for choices, the combo box will display five options combined from the two. If any of the three options specified in approvedChoices is selected, the group is considered approved. In this case, even if the elements Diff or Reject belong to a group, and if "Show diff only" is enabled, corresponding group will not appear in the group table.

The following is a sample of group flag definition file.

Sample 1(Default setting)

```
[{"name": "Approved flag", "choices": ["Reject"], "approvedChoices": ["OK"]}]
```

Sample 2

```
[
  {
    "name": "quality",
    "choices": [
      "A",
      "B",
      "C"
    ]
  },
  {
    "name": "confirm result",
    "choices": [
      "Reject",
      "n/a"
    ],
    "approvedChoices": [
      "Fair",
      "Good",
      "Excellent",
    ]
  },
  {
    "name": "Expected User",
    "choices": [
      "UserA",
      "UserB",
    ]
  },
]
```

III. About geometry comparison log

Below shows the forms and corresponding explanations when exporting XML log.

<CompareResultSummary>: Summary of the comparison result

1. title="numOfSourceEnfBody"
2. title="numOfTargetEnfBody"
3. title="numOfSourceRemovedPolygonComponent"
4. title="numOfTargetRemovedPolygonComponent"
5. title="numOfComparedBody(Source+Target)"
6. title="numOfBodyWithEdgeDistanceDifference"
7. title="numOfBodyWithFaceDistanceDifference"
8. title="numOfBodyWithContinuityDifference"
9. title="numOfSourceBodyWithNoCorrespondenceEdge"
10. title="numOfSourceBodyWithNoCorrespondenceFace"
11. title="numOfTargetBodyWithNoCorrespondenceEdge"
12. title="numOfTargetBodyWithNoCorrespondenceFace"
13. title="numOfBodyWithUniqueFace"
14. title="numOfDiffFailedBody"
15. title="numOfTreeNodeWithoutPair"
16. title="numOfTreeNodePairWithNameDiff"
17. title="numOfTreeNodePairWithPartNameDiff"
18. title="numOfTreeNodePairWithPartNumberDiff"
19. title="numOfTreeNodePairWithInstPathMatrixDiff"
20. title="numOfTreeNodePairWithParentAssemblyDiff"
21. title="numOfTreeNodePairWithGeometryDiff"
22. title="numOfDiffFailedEdge"
23. title="numOfDiffFailedFace"
24. title="numOfFaceGeometryDiff"
25. title="numOfIsoEdgeGeometryDiff"
26. title="numOfFreeEdgeGeometryDiff"
27. title="numOfIsoVertexGeometryDiff"
28. title="numOfFaceNormalDiff"
29. title="numOfPolygonDiff"
30. title="numOfPolylineDiff"
31. title="numOfPolygonPointDiff"
32. title="numOfSystemPropertyDiff"
33. title="numOfUserPropertyDiff"
34. title="numOfBrepAttrFaceDiff"
35. title="numOfBrepAttrIsoEdgeDiff"
36. title="numOfBrepAttrIsoVertexDiff"
37. title="numOfPMINoteDiff"
38. title="numOfPMIDatumDiff"
39. title="numOfPMIDatumTargetDiff"
40. title="numOfPMIRoughnessDiff"
41. title="numOfPMIGDTHDiff"
42. title="numOfPMIDimensionDiff"
43. title="numOfPMILocatorDiff"
44. title="numOfPMISpotWeldDiff"
45. title="numOfPMILineWeldDiff"
46. title="numOfPMICameraDiff"

1. Total number of parts in the first model specified with <sourcefile>
2. Total number of parts in the second model specified with <targetfile>
3. Total number of polygon parts deleted to run Detailed / Quick Comparison normally, in the first model specified with <sourcefile>
4. Total number of polygon parts deleted to run Detailed / Quick Comparison normally, in the second model specified with <targetfile>
5. The compared part number (an even number because it counts both separately)
6. The total part number in source file (ENF 1) that includes more than 1 edge exceeds the minimum edge distance value as the result of detailed comparison. (Valid only when running the option CheckEdge = 1 and ComparisonMode = 0)
7. The total part number in source file (ENF 1) that includes more than 1 face exceeds the minimum face distance value as the result of detailed comparison. (Valid only when running the option CheckFace = 1 and ComparisonMode = 0)
8. The total part number in source file (ENF 1) that includes more than 1 edge exceeds the minimum edge continuity value as the result of comparison. (Valid only when running the option CheckCont = 1 and ComparisonMode = 0)
9. The total part number in source file (ENF 1) that does not include corresponding edge as the result of topological comparison. (Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
10. The total part number in source file (ENF 1) that does not include corresponding face as the result of topological comparison. (Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
11. The total part number in compared file (ENF 2) that does not include corresponding edge as the result of topological comparison. (Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
12. The total part number in compared file (ENF 2) that does not include corresponding face as the result of topological comparison. (Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
13. The total part number that includes UniqueFace. (Valid only when running the option ComparisonMode = 1)
14. The total part number that includes more than one failure of comparison.
15. The number of components with no paired element as a result of structure comparison.
16. The number of pair of elements whose names are different from each other as a result of structure comparison.
17. The number of pair of elements whose property "PartName" are different from each other as a result of structure comparison.
18. The number of pair of elements whose property "PartNumber" are different from each other as a result of structure comparison.

19. The number of pair of elements whose instance path matrices are different as a result of structure comparison.
20. The number of pair of elements whose parent assemblies are not paired as a result of structure comparison.
21. The number of pair of elements whose geometries are different as a result of structure comparison.
22. The number of edges on which "Comparison for Quick Viewing" has been failed.
23. The number of faces on which "Comparison for Quick Viewing" has been failed.
24. The number of face differences detected in "Comparison for Quick Viewing."
25. The number of isolated curve differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckEdge = 1)
26. The number of free edge differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckEdge = 1)
27. The number of isolated point differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckVertex = 1)
28. The number of face normal differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckAngleFaceNormal = 1)
29. The number of differences between exact geometry and tessellated geometry detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPolygon = 1)
30. The number of polyline differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPolyline = 1)
31. The number of polygon point differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPolygonPoint = 1)
32. The number of system property differences detected in "Comparison for Quick Viewing." (Valid only when running the option CheckSystemProperty = 1)
33. The number of user property differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckUserProperty = 1)
34. The number of face attribute differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckAttributeFace = 1)
35. The number of isolated curve attribute differences detected in "Comparison for Quick Viewing." (Valid only when running the option CheckAttributeIsolatedEdge = 1)
36. The number of isolated point attribute differences detected in "Comparison for Quick Viewing." (Valid only when running the option CheckAttributeIsolatedVertex = 1)
37. The number of note differences detected in "Comparison for Quick Viewing." (Valid only when running the option CheckPMI = 1)

38. The number of datum differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
39. The number of datum target differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
40. The number of surface finish differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
41. The number of GDT differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
42. The number of dimension differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
43. The number of locator differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
44. The number of spot weld differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
45. The number of line weld differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)
46. The number of model view differences detected in "Comparison for Quick Viewing."
(Valid only when running the option CheckPMI = 1)

<GeomDiff>: Comparison in entity distance

1. <DiffEdge dist="outOfRange">
 <DekEntity entityType="Edge" bodyDekId="2" dekId="17" param="0.125000" />
 <SourceLocalPoint x="20.000000" y="10.000000" z="0.000000" />
 </DiffEdge>
2. <DiffEdge dist="0.165817">
 <DekEntity entityType="Edge" bodyDekId="2" dekId="35" param="0.500000" />
 <DekEntity entityType="Edge" bodyDekId="3" dekId="32" param="0.500007" />
 <SourceLocalPoint x="25.4434" y="2.5129" z="-0.00051845" />
 </DiffEdge>
3. <DiffEdgeNum number="20" />
4. <DiffFace dist="outOfRange">
 <DekEntity entityType="Face" bodyDekId="2" dekId="1" param="0.285714,
 0.216701" />
 <SourceLocalPoint x="6.574629" y="-29.982826" z="12.699250" />
 </DiffFace>

```

5. <DiffFace dist="0.457712">
    <DekEntity entityType="Face" bodyDekId="2" dekId="13" param="1.000000,
0.500912" />
    <DekEntity entityType="Face" bodyDekId="3" dekId="11" param="0.843355,
0.639853" />
    <SourceLocalPoint x="25.725615" y="-20.942516" z="2.470861" />
    <TargetLocalPoint x="25.668945" y="-20.787205" z="2.580616" />
</DiffFace>

6. <DiffFaceNum number="10" />

```

1. Part ID and Edge ID of the edge that exceeds maximum deviation distance and also the location of gap occurred.
The coordinate value of a corresponding edge in a part. (typical one point)
(Valid only when running the option CheckEdge = 1 and ComparisonMode = 0)
2. PartID and EdgeID of the edge detected that has the gap within the threshold. The second DekEntity is the distance against the first DekEntity. (The distance between these two points is "dist".)
The coordinate value of a corresponding edge in a part. (typical one point)
(Valid only when running the option CheckEdge = 1 and ComparisonMode = 0)
3. The sum of all edges detected as Edge Distance in the model in left view.
(Valid only when running the option CheckEdge = 1 and ComparisonMode = 0)
4. PartID and FaceID of the face that exceeds the maximum deviation distance and also the location of gap occurred.
The coordinate value of a corresponding face in a part. (typical one point)
(Valid only when running the option CheckFace = 1 and ComparisonMode = 0)
5. PartID and FaceID of the face detected that has the gap within the threshold. The second DekEntity is the distance against the first DekEntity. (The distance between these two points is "dist".)
The coordinate value of a corresponding face in a part. (typical one point)
(Valid only when running the option CheckFace = 1 and ComparisonMode = 0)
6. The sum of all faces detected as Face Distance in the model in left view.
(Valid only when running the option CheckFace = 1 and ComparisonMode = 0)

<BodyIdToComponentIdMap>: correspondence of body id and component id.

```
1. <Pair bodyId="1" componentId="1" />
```

1. It shows the correspondence of the part (id=1) and Component (id=1).

<ElementCorre>: Corresponding Elements

1. <ElementCorreGrp elementCorreType="diffFailFace" elementCorreID="8">
 <CorreGrpElement bodyDekId="2" dekId="6" />
 <SourceLocalPoint x="6.000000" y="1.500000" z="0.000000" />
 <SourceLocalPoint x="-10.500000" y="18.000000" z="-0.000000" />
 <SourceLocalPoint x="-53.580651" y="7.258787" z="0.000000" />
 <SourceLocalPoint x="-60.403005" y="-15.056129" z="0.000000" />
 <Error Value="213" description="Face geometry comparison error. This face
 doesn't have closed UV-polyline loop to define face region." />
 </ElementCorreGrp>
2. <ElementCorreGrp elementCorreType="sourceBody" elementCorreID="2"> +
 <CorreGrpElement bodyDekId="1" dekId="1" />
 </ElementCorreGrp>
3. <ElementCorreGrp elementCorreType="correspondenceBody1to1" elementCorreID="1">
 <CorreGrpElement bodyDekId="2" dekId="2" />
 </ElementCorreGrp>
4. <ElementCorreGrp elementCorreType="correspondenceEdge1to1" elementCorreID="8">
 <CorreGrpElement bodyDekId="2" dekId="6" />
 </ElementCorreGrp>
5. <ElementCorreGrp elementCorreType="correspondenceFace1to1" elementCorreID="43">
 <CorreGrpElement bodyDekId="2" dekId="17" />
 </ElementCorreGrp>
6. <ElementCorreGrp elementCorreType="correspondenceFaceMtoN" elementCorreID="55">
 <CorreGrpElement bodyDekId="2" dekId="12" />
 <CorreGrpElement bodyDekId="2" dekId="10" />
 </ElementCorreGrp>
7. <NoCorreEdge bodyDekId="1" dekId="4" />
8. <NoCorreFace bodyDekId="2" dekId="1" />
9. <NoCorreEdgeNum number="9" />
10. <NoCorreFaceNum number="2" />
11. <UniqueFaceGroupOfBodyNum bodyDekId="1" number="1" />
12. <UniqueFaceGroupNum number="4" />

1. It shows the comparison of the face (ID=6) existed in the part (ID=2) has been failed.
Local coordinate points of the vertices to identify a failed face.
The description of the reason of comparison failure.
2. It shows the part (ID=1) is the source model. (bodyDekId and dekId is always the same)
3. It shows the comparing part (ID=2) exists in the part with included. (bodyDekId and dekId is always the same.)
One more ElementCorreGrp exists that has the same elementCorreID value and those are compared to each other.
4. It shows that the edge (PartID=2, EdgeID=6) has a corresponding edge. One more ElementCorreGrp exists that has same value of elementCorreID and it corresponds with another.
(Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
5. It shows the face (PartID=2, FaceID=17) has a corresponding face. One more ElementCorreGrp exists that has same value of elementCorreID and it is corresponding with it.
(Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
6. It shows that 2 faces (PartID=2, FaceID=12 and FaceID=10) have corresponding face as M:N relations.
One more ElementCorreGrp with the same value of elementCorreID exists and corresponds with the face group.
(Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
(There is no example of edge but correspondenceEdgeMtoN is work as same as face)
7. It shows the there is no corresponding entity as 1:1, either M:N relation with the edge (PartID=1, EdgeID=4)
(Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
8. It shows the Face (PartID=2) does not have corresponding entity as 1:1, either M:N relation.
(Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
9. Sum of edges that does not have corresponding entity as 1:1, either M:N. (Original Model+Comparing model)
(Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
10. Sum of faces that does not have corresponding entity as 1:1, either M:N. (Original Model+Comparing model)
(Valid only when running the option CheckTopology = 1 and ComparisonMode = 0)
11. Sum of faces that does not have corresponding entity when ComparisonMode=1 is set. (Original Model+Comparing model)
(Valid only when running the option ComparisonMode = 1)
12. Sum of face groups that does not have corresponding entity when ComparisonMode=1 is set. (Original Model+Comparing model)
(Valid only when running the option ComparisonMode = 1)

<ContDiff>: Continuity

```

1. <ContDiffPair contDiffType="2" param="0.062500" angle="1.611273">
    <ContDiffElem bodyDekId="2" dekId="8" />
    <ContDiffElem bodyDekId="3" dekId="8" />
</ContDiffPair>

2. <ContDiffPair contDiffType="3" param="0.374972" angle="0.619772">
    <ContDiffElem bodyDekId="2" dekId="25" />
    <ContDiffElem bodyDekId="3" dekId="25" />
</ContDiffPair>

3. <ContDiffNum number="3" />

```

1. The contDiffType=2 shows it caused creased geometry in the target model against the area that has creased geometry within the threshold.
The first ContDiffElem indicates the EdgeID=8 in the source model, the second ContDiffElem indicates the EdgeID=8 in the target model.
Parameter and creased angle states about the edge in source model.
(Valid only when running the option CheckCont = 1 and ComparisonMode = 0)
2. The contDiffType=3 shows the creased area more than the limitation in the target model was repaired in source model.
The first ContDiffElem indicates the EdgeID=25 of left model. The second ContDiffElem indicates the EdgeID=25 of target model.
Parameter and creased angle are about the edge on target model.
(Valid only when running the option CheckCont = 1 and ComparisonMode = 0)
3. The number of changes in continuous edges. (generated/repaired creases)
(Valid only when running the option CheckCont = 1 and ComparisonMode = 0)

IV. Restriction

Please note that you cannot install CAD Validator using UNC paths.

V. How to Customize (Customization File)

You can tune up the validation settings for "Comparison for Quick Viewing" as follows;

- **Settings on Geometry Validation**
Adjust the settings in parameter files.
- **Settings on PMI / Attribute / B-rep Attribute Validation**
Change the followings in customization files ("customize_table.csv" and "customize_utility.rb"). Please see below for the instruction.
 - Add/Delete elements to compare

- Customize the part/assembly mapping logic
- Customize the validation logic

To customize settings on PMI, attribute and B-rep attribute validation,

1. Prepare customization files ("customize_table.csv" and "customize_utility.rb") as described below.
2. Specify them by their folder path with "CustomizeFolder" parameter as;
CustomizeFolder=<Absolute folder path>

customize_table.csv

This is a configuration file in table format to allow following customizations;

- Add/Delete elements and their properties (belonging data such as name, visibility etc.) to validate
- Change the part/assembly mapping logic
- Apply a different validation logic for each property
- Apply a different validation tolerance for each property

customize_utility.rb

This is an additional script file (detailed process is written in this file, and "customize_table.csv" refers to this file) for an advanced customization of the mapping and validation logic. This features high maintainability compared to the customization using "customize_table.csv" only, and is effective especially when the logic is complex.

Other than the customization using "customize_table.csv", you can also add/delete elements to validate by enabling/disabling the following parameters.

- CheckInstance
- CheckPMI
- CheckSystemProperty
- CheckUserProperty
- CheckAttributeFace
- CheckAttributeIsolatedEdge
- CheckAttributeIsolatedVertex

Recommended settings differ from one CAD format to another, and sample "customize_table.csv" files and "customize_utility.rb" files with recommended standard settings are available for major CAD systems under the following folder.

- <ASFALIS Components>\tools\win\cadvalidator\template\

Please contact to Elysium or its authorized resellers (your primary contact on ASFALIS-related

inquiries) if you wish to customize further than modifying the value of sample customization files.

VI. Report Files

You can also export the validation result in reports in the following formats. It requires no licensing to view the report at receivers' end.

Report Type	2D XML	3D HTML	3D PDF	Diff-Analyzer
Visualization - Geometry	✓	✓	✓	✓
Visualization - PMI	✗	✓	✓	✗
Visualization - Attribute	✗	✓	✓	✗
Performance - Large Assembly	✗	✓	▲	✓
Dynamic Report	✗	✓	✓	✓
Comment Insertion	✗	✗	✓	✓
Output File Type	Multiple files	Multiple files	Single file	Single file
Prerequisites	Internet Explorer 11	Internet Explorer 11 (*1)	Acrobat Reader DC or later	64-bit OS, Windows 8.1 or later (*2)

(*1): It requires ActiveX control enabled.

(*2): Output report (exe file) is not digitally signed, and it may require administrative privileges to open the report depending on the OS security settings.

2D XML Validation Report

You can make a report (XML) automatically from comparison results.

1. How to generate

Specify the parameter CreateReport=1.

A report file name can be specified by xml scenario tag <xmlreportfile>.

If the parameter CreateReport=1 is specified and the tag <xmlreportfile> is not specified, the report file name will be automatically determined based on the name of the <diffresultfile> and <xmllogfile>.

- i. A file name of the tag <diffresultfile> plus "_report.xml" if the tag <diffresultfile> is specified.
- ii. A file name of the tag <xmllogfile> plus "_report.xml", if the tag <diffresultfile> is not specified.

2. Contents of output files

In addition to the xml file that is specified in the tag <xmlreportfile>, auxiliary files will be created in the same folder. In case of copying the result report to other locations, sending over to colleagues etc., please ensure to copy the entire folder including all the followings. EX) Output report file name is "ReportFileName.xml"

ReportFileName.xml	Validation report file
ReportFileName	Folder which contains captured images and viewpoint files. (The folder name is the same as the report name)
elyReportComp***.xsl	Style sheet to convert 2D XML report to HTML for visualization on browsers.

3. About View Point File

A view point file will be exported as well as the comparison result file after the geometry comparison (detailed comparison or quick comparison). Drag and drop the view point file to Model Viewer or CADdoctor, and then the model will be displayed with the same view point as the image in the comparison result file.

Please note that the view point files will not be exported for comparison for quick viewing.

4. Behavior and Restriction

- Please note that the CPU and memory consumption of report generation process may get greater than that of validation process depending on the target models.
- Validation result files specified with "diffresultfile" tag and "xmllogfile" tag (these are created during the validation process) will be available even when it fails to generate validation reports due to performance issues, etc. because the report generation process starts upon the completion of the validation process.
- Please note that the validation report generation will be skipped in case of name conflicts; there are XML file(s) and/or image folder(s) whose filename(s) or folder name(s) conflict with the specified name.
- Please open the validation reports with Internet Explorer 11. Other browsers are not supported.
- Please ensure to specify a different folder as the output folder for XML and 3D HTML validation reports. (You cannot specify the same file path with different file extensions to parameter "xmlreportfile" and "html3dreportfile".) This is because a subfolder will be created for each format (XML and 3D HTML) in the output folder under the same name as the report file.)
- Please note that the report will not be exported when there is a file or a folder under the same file path (without file extension) as what you specified with "xmlreportfile".

3D HTML Validation Report

1. How to generate

Specify the parameter "Create3DReport=1" to export a result report in 3D HTML file.

You can specify the report filename in the XML Scenario file using the tag `<html3dreportfile>`, or the filename will be automatically determined based on the value of the `<diffresultfile>` and `<xmllogfile>` when the tag `<html3dreportfile>` is not specified.

- i. A file name of the tag `<diffresultfile>` plus `"_3dreport.html"` if the tag `<diffresultfile>` is specified.
- ii. A file name of the tag `<xmllogfile>` plus `"_3dreport.html"` if the tag `<diffresultfile>` is not specified.

2. Contents of output files

In addition to the report file in HTML format that is specified using the tag `<html3dreportfile>`, auxiliary files will be created in the same folder.

In case that a report file name is `ReportFileName.html`, output files are as below:

<code>ReportFileName.html</code>	The top file of report
<code>ReportFileName</code>	Report file, JavaScript, CSS, result data of comparison will be saved in the folder. (The folder name is the same as the report name)

Please ensure to copy including the folder `"ReportFileName"` when moving or creating a copy of the report file (`ReportFileName.html`).

3. Behavior and Restriction

- Please open the validation reports with Internet Explorer 11. Other browsers are not supported.
- Specify the following parameter to view the exported report in the environment without internet connection.
`"Enclose3DReportAddon=1"`
 Then, run `"RegisterFlgControlGV.bat"` placed in the subfolder for 3D HTML report as Administrator in the computer to open the report.
- Please ensure to specify a different folder as the output folder for XML and 3D HTML validation reports. (You cannot specify the same file path with different file extensions to parameter `"xmlreportfile"` and `"html3dreportfile"`.) This is because a subfolder will be created for each format (XML and 3D HTML) in the output folder under the same name as the report file.)
- Please note that the report will not be exported when there is a file or a folder under the same file path (without file extension) as what you specified with `"html3dreportfile"`.

3D PDF Validation Report

1. How to generate

Specify the parameter `"Create3DPdfReport=1"` to export a result report in 3D PDF file.

You can specify the report filename in the XML Scenario file using the tag `<pdf3dreportfile>`, or the filename will be automatically determined based on the value of the `<diffresultfile>` and `<xmllogfile>` when the tag `<pdf3dreportfile>` is not specified.

- i. A file name of the tag <diffresultfile> plus "_3dpdfreport.pdf" if the tag <diffresultfile> is specified.
- ii. A file name of the tag <xmllogfile> plus "_3dpdfreport.pdf" if the tag <diffresultfile> is not specified.

2. Behavior and Restriction

- Please note that the report will not be exported when there is a file under the same file path as what you specified with "pdf3dreportfile".
- The process of creating report is heavier than that of comparison.
- Please also check [3.3.9, "3D PDF Editor"](#) when using this Component in combination with PDF Editor Component.

Diff-Analyzer Report File

1. How to generate

Specify the parameter "CreateDiffAnalyzer=1" to export a result report as Diff-Analyzer, a self-contained visualization file (exe file).

You can specify the report filename in the XML Scenario file using the tag <diffanalyzer>, or the filename will be automatically determined as follows based on the value of the <diffresultfile> or <xmllogfile> when the tag <diffanalyzer> is not specified.

- i. When the tag <diffresultfile> is specified:
"<Filename specified with <diffresultfile> tag>_diffanalyzer.exe"
- ii. When the tag <diffresultfile> is not specified:
"<Filename specified with <xmllogfile> tag>_diffanalyzer.exe"

2. Behavior and Restriction

- Functionality is equivalent to that of "Diff-Viewer", but the viewable result is limited to what the report was exported of.
- Diff-Analyzer is a self-contained visualization file, and requires no installation nor licensing at receiver's side.
- Because the report generation process starts upon the completion of the validation process, the validation result files specified with "diffresultfile" tag and "xmllogfile" tag (these are created during the validation process) will be available even when it fails to generation validation reports due to performance issues, etc.
- Please note that the report will not be exported when there is a file under the same file path as what you specified with "diffanalyzer".
- Hardware requirements to open Diff-Analyzer is equivalent to that of Model Viewer. Please refer to the release note included in the installation package.
 - <ASFALIS installation package>base\doc\ASFALIS_Model_Viewer

How to Use

Please refer to "Diff-Viewer Help" for details about the functionalities. Help file is included in the Diff-Analyzer, too. It is available from the [?] icon at top right corner.

3.3.4. Attribute Editor

It is a component that operates attribute information in neutral file. It is possible to execute the following operations:

- [Edit attribute](#)
- [Extract attribute](#)
- [Replace attribute value](#)
- [Add attribute value](#)
- [Editing instance name](#)

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, “XML Scenario”](#).

- XML scenario sample

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScenarioList>
  <Scenario>
    <!--Convert CAD to neutral file -->
    <CAD2ENF>
      <inputfile path="${INPUTFILE}" />
      <outputfile path="${OUTPUTNAME}_a.enf"/>
      <parameterfile path="${PARAMETERFILE}"/>
      <productcode id="${PRODUCTID}"/>
      <workdir path="${WORKDIR}"/>
      <logfile path="${OUTPUTNAME}.log"/>
      <xmllogfile path="${OUTPUTNAME}.xml"/>
    </CAD2ENF>

    <!--Attribute mapping -->
    <ATTREDIT>
      <inputfile path="${INPUTNAME}_a.enf"/>      <!--Input file -->
      <outputfile path="${OUTPUTNAME}_b.enf"/>    <!--outputfile -->
      <mapfile path="mapping.xml" />              <!--mappingfile -->
      <bomfile path="bom.xml" />                   <!--attribute outputfile -->
      <logfile path="${OUTPUTNAME}_b.log"/>        <!--logfile -->
      <workdir path="${WORKDIR}"/>
    </ATTREDIT>
  </Scenario>
</ScenarioList>
```

II. Editing attribute

The following operations are possible. The following operation is individually described

respectively as "Rule". It is possible to specify two or more "Rule" at the same time.

1. Create (type = "create")

Convert the value of attribute from "FROM" to "TO".

- Format

Attribute key and type of "From" and "To" are defined by one rule

Type is specified by either character string of "System" or "User"

2. Delete (type = "delete")

Delete the specified attributes.

- Format

Enumerate the attributes to be deleted

3. Combine (type = "combine")

Connect the specified attributes with the character string, and create a new attribute.

- Format

Arrange attributes that unites and character string in "From" tag.

You can select items from attribute, string and Serial Number.

"Serial Number" is a number that is incremented by 1 as attributes get created. (Initial value is 0.)

- Attribute: Target attribute
- String: String which is combined with value attribute
- Serial Number: Specify digit number by using "digit" attribute

By specifying the "target" attribute for each Rule, you can specify the object element of the operation.

- Sample

```
<?xml version="1.0" encoding="Shift_JIS"?>
<AttrEdit>
  <MappingRules>
    <Rule type="create">
      <From key="Author" type="user"/>
      <To key="Creator" type="user"/>
    </Rule>
    <Rule type="delete">
      <Attribute key="Title" type="user"/>
      <Attribute key="Author" type="user"/>
    </Rule>
    <Rule type="combine" target="assembly">
      <From>
        <Attribute key="Name" type="system"/>
        <String value="_"/>
        <Attribute key="PartNumber" type="system"/>
      </From>
      <To key="Name" type="system"/>
    </Rule>
    <Rule type="combine" target="part">
      <From>
        <Attribute key="Name" type="system"/>
        <String value="_"/>
        <SerialNumber digit="4"/>
      </From>
      <To key="Name" type="system"/>
    </Rule>
  </MappingRules>
</AttrEdit>
```

- The Rule will be ignored when the format is incorrect in the Rule tag.
- Only the first rule will be used and all the rest will be ignored when there exists more than one rule in which the type of the target attribute duplicates each other. Warning message will be written in the log file, too
- You must describe values according to the XML format. When you use the following characters in XML file, you must describe the string as follows.

```
'&':&amp; ' '>':&gt; ' '<':&lt; ' "'':&quot; ' "'':&apos;
e.g.,) not available: <String value="Name&Number"/>
available: <String value="Name&amp;Number"/>
```



- When you use "force" attributes and specify "on" for that in the "Rule" tag, attributes will be edited even if the value of the target attribute is empty. e.g.,)

```
<Rule type="combine" force="on">
```

- Only the attributes with the following keys will be regarded as "System" properties.

Name	PartName	PartNumber
NFName	ConfigName	BinName
ChangeHistory	Description	PartComment
PartDefinition	PartVersion	PartRevision
PartSource	TFName	PartLayer

III. Extracting attribute

Attribute information included in neutral file can be written by the XML form as another file. The output file is specified for "bomfile" in the XML scenario.

- Sample of the BOM file


```
<?xml version="1.0" encoding="UTF-8"?>
<AttrList>
  <Owner name="bom_sample_out" type="model">
  </Owner>
  <Owner name="top_asm_sample" type="assembly">
    <Attr class="system" key="PartNumber" type="text">sample</Attr>
  </Owner>
  <Owner name="Asm1_01" type="assembly">
    <Attr class="system" key="PartNumber" type="text">01</Attr>
    <Attr class="system" key="PartRevision" type="text">12345</Attr>
  </Owner>
  <Owner name="part1" type="part">
    <Attr class="system" key="PartName" type="text">part1</Attr>
    <Attr class="system" key="PartNumber" type="text">part1_number</Attr>
  </Owner>
  <Owner name="Asm2_02" type="assembly">
    <Attr class="system" key="PartNumber" type="text">02</Attr>
    <Attr class="system" key="PartRevision" type="text">67890</Attr>
  </Owner>
  <Owner name="part2" type="part">
    <Attr class="system" key="PartName" type="text">Part2</Attr>
  </Owner>
  <Owner name="part3" type="part">
    <Attr class="system" key="Description" type="text">UserAttribute</Attr>
    <Attr class="system" key="Material" type="text">Default</Attr>
    <Attr class="system" key="PartName" type="text">part3</Attr>
    <Attr class="user" key="Creator" type="text">yamada</Attr>
    <Attr class="user" key="Subject" type="text">test_model</Attr>
  </Owner>
</AttrList>
```

IV. Replacing attribute value

You can replace attribute value by using ValueReplacementRules.

<ValueReplacementRules> tag

You need to write the setting of this function between ValueReplacementRules tags in AttrEdit tag. You can write multiple Rule tags, too.

<Rule> tag

Following options can be set as the attribute. (Option)

target: Specify the element type from "part," "assembly" or "model." Only the attributes of specified element types will be replaced. All attributes will be replaced in case you don't specify the target attributes.

regex: Specify "on" and also the "search and replace" values so that "search and/or replace" strings become regular expressions. Specify "off" or write nothing, then "search and replace"

value will not be regular expressions. (The exact value specified at "search" will be replaced to the value at "replace.")

Specify the target attribute (the key and the type) and the character string to replace from and to between the "Rule" tag.

Only the first rule will be used and all the rest will be ignored when there exists more than one rule in which the type of the target attribute duplicates each other. Warning message will be written in the log file, too.

<Attribute> tag

Specify target attribute by using the following attributes. (Mandatory)

key: Specify the attribute key.

type: Specify the attribute type, system property or user property by "system" or "user."

<Value> tag

Specify the value of the attribute to replace from and to. (Mandatory)

from: Specify the search string.

to: Specify the replacement string.

Write "Value" tag right after "Attribute" tag. You can specify multiple tags, too. All the strings which match the specified value will be replaced. Replacement will be performed from the first rule checking through all the attributes before moving on to the next rule. Once a string has been replaced, it will not be replaced again even when the string matches the search string in the tag(s) written below.

- Example

```
<?xml version="1.0"?>
<AttrEdit>
  <ValueReplacementRules>
    <Rule target="part" regexp="off">
      <Attribute key="Material" type="system"/>
      <Value from="Steel" to="Metal"/>
      <Value from="Copper" to="Bronze"/>
    </Rule>
    <Rule regexp="on">
      <Attribute key="Name" type="system"/>
      <Value from="^AAA.*" to="ABC"/>
    </Rule>
  </ValueReplacementRules>
</AttrEdit>
```

With the mapping rule above, regarding the "Material" property which is a system property, the value will be replaced from "Steel" to "Metal" and "Copper" to "Bronze."

And regarding the "Name" property which is also a system property, all the values beginning with "AAA" will be replaced to "ABC".

V. Adding attribute value

You can set attribute value by using ValueAdditionRules.

<ValueAdditionRules> tag

You need to write the setting of this function between ValueAdditionRules tags in AttrEdit tag. You can write multiple Rule tags, too.

Following options can be set as the attribute. (Option)

overwrite: Specify "on" or "off." When "on" is specified, the value will be overwritten even if an attribute under the same name already exists. It will be regarded as "off" when the attribute "overwrite" is not specified. A system property and a user property will be regarded as the different attributes even if they have the same attribute key.

<Rule> tag

Following options can be set as the attribute. (Option)

target: Specify the element type from "part," "assembly" or "model." Attributes will be added only to the attributes of specified element types.

You need to write the name of the attribute to add attributes to, and the attribute value to add between "Rule" tags.

<TargetName> tag

Specify the name of the target part or assembly as the value attribute of the tag. The attribute will be added only when the name exactly matches the specified value. All elements will be the target when this tag is not specified. The attribute will be added to the model regardless whether "TargetName" tag is specified or what is specified for "value" attribute, when the target="model" is specified in the Rule tag.

Following options can be set as the attribute.

regex: Specify "on" and also the "search" value so that "search" string becomes a regular expression. Specify "off" or write nothing, then "search" value will not be regular expressions.

<Attribute> tag

Specify target attribute by using the following attributes. (Mandatory)

key: Specify the attribute key.

type: Specify the attribute type, system property or user property by "system" or "user."

<Value> tag

Specify the value to be added of the attribute. You need to write the attribute value to add between "Value" tags. (Mandatory)

- Example

```
<?xml version="1.0"?>
<AttrEdit>
  <ValueAdditionRules overwrite="on">
    <Rule target="part">
      <TargetName regexp="on" value="^Arm200-101-\d*$"/>
      <Attribute key="Material" type="system"/>
      <Value>copper</Value>
    </Rule>
    <Rule target="assembly">
      <TargetName regexp="on" value="^Arm200-asm-\d*$"/>
      <Attribute key="Maker" type="user"/>
      <Value>ABC Corp.</Value>
    </Rule>
    <Rule target="model">
      <Attribute key="Designer" type="user"/>
      <Value> Yamada Taro</Value>
    </Rule>
  </ValueAdditionRules>
</AttrEdit>
```

With the mapping rule above, the attribute "Material" will be added to the parts whose name begin with "Arm200-101," and the attribute "Maker" will be added to the assemblies whose name begin with "Arm200-asm-."

Also the attribute "Designer" will be added to the model.

VI. Editing instance name

You can edit instance name by using TargetSpecificRules.

<TargetSpecificRules> tag

You need to write the setting of this function between TargetSpecificRules tags in AttrEdit tag.

<Rule> tag

Specify following options as the attribute.

type="combine"

target="instance"

target: "instance" is effective only when editing instance name.

<From> tag

Specify the items you want to connect in the "From" node.

You can select items from attribute, string and Serial Number.

<String> tag

value: Specify the character string you want to use for connecting.

<SerialNumber> tag

digit: Specify digit number (selectable range: 1-20, omissible).

<Attribute> tag

Specify following options as the attribute.

source: "Parent" or "Child" source

"Parent" means the upper assembly of the instance.

"Child" means the child part or assembly of the instance.

key: Specify the attribute key of the element that specified by "source".

type: Specify the "system" attribute or "user" attribute.

<To> tag

Specify following options as the attribute.

key="Name"

- **Example**

You can get the instance name (with sequence number) by combining parent assembly name and child part/assembly name via "_" (underscore).

```
<AttrEdit>
  <TargetSpecificRules>
    <Rule type="combine" target="instance">
      <From>
        <Attribute source="Parent" key="Name" type="system"/>
        <String value="_"/>
        <Attribute source="Child" key="Name" type="system"/>
        <SerialNumber digit="1"/>
      </From>
      <To key="Name"/>
    </Rule>
  </TargetSpecificRules>
</AttrEdit>
```

3.3.5. Assembly Editor

By using this component, you can change assembly structure and delete specified part/assembly.

- [Edit assembly](#)
- [Delete specified part and assembly](#)
- [Extract sub-assembly and part from assembly](#)

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, "XML Scenario"](#).

- XML scenario sample (Deleting part/assembly)

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScenarioList>
  <Scenario>
    <CAD2ENF>
      <inputfile path="{INPUTFILE}"/>
      <outputfile path="{OUTPUTNAME}_a.enf"/>
      <parameterfile path="{PARAMETERFILE}"/>
      <productcode id="{PRODUCTID}"/>
      <workdir path="{WORKDIR}"/>
      <logfile path="{OUTPUTNAME}.log"/>
      <xmllogfile path="{OUTPUTNAME}.xml"/>
    </CAD2ENF>

    <ASMEDIT>
      <inputfile path="{OUTPUTNAME}_a.enf"/>
      <outputfile path="{OUTPUTNAME}_b.enf"/>
      <parameterfile path="{PARAMETERFILE}"/>
      <namelist path="deleteList.txt"/>
      <workdir path="{WORKDIR}"/>
      <logfile path="{OUTPUTNAME}_b.log"/>
    </ASMEDIT>
  </Scenario>
</ScenarioList>
```

- XML scenario sample (Extracting sub-assembly/part from assembly)

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScenarioList>
  <Scenario>
    <ASMEDIT>
      <inputfile path="{INPUTNAME}.enf"/>           <!-- input file -->
      <outputfile path="{OUTPUTNAME}.enf"/>         <!-- output file -->
      <parameterfile path="{PARAMETERFILE}"/>       <!-- parameter file -->
      <extraction_target path="target.xml"/>         <!-- extract target list -->
      <extraction_outputdir path="{OUTPUTPATH}\extraction"/>
                                                    <!-- output folder for extraction -->
      <workdir path="{WORKDIR}"/>                   <!-- Working directory -->
      <logfile path="{INPUTNAME}_extraction.log"/>   <!-- Log file -->
    </ASMEDIT>
  </Scenario>
</ScenarioList>
```

II. Deleting part/assembly

You can delete part or assembly by using delete file list.

- Format of the namelist file.
 - The namelist file is a text file which lists the name of the parts and the assemblies to delete. The names are delimited by CR/LF code. If UseRegexpInNameList=1 is specified, the listed names will be considered as strings using regular expression.
 - If DeleteSpecifiedPart=1 is specified, a part will be deleted when its name exactly matches one of the names in the namelist. And assemblies, if DeleteSpecifiedAssembly=1 is specified, an assembly will be deleted when its name exactly matches one of the names in the namelist.
- Specification
 - When there exists more than one component under the same name which matches one of the names in the namelist, all of them will be deleted.
 - The parent assembly will remain (will not be deleted) even if it becomes empty after deleting the components underneath.
 - When deleting an assembly, the child parts and child assemblies underneath will also be deleted. In case those child parts or assemblies are shared by other assemblies, only that instance will be deleted and the shared parts or assemblies at other instances will remain.
 - If the namelist file is not specified, does not exist or cannot be read, the process afterward will not be executed, and the program will be closed. It will be written as "Failed to open list" in the log file.
 - If the component to delete which is listed in the namelist does not exist, it will be skipped and move onto the next one. It will be written as "Component XXX was not found in the model" in the log file.

III. Editing Assembly

You can edit assembly structure by using this component. For detail, please refer parameter manual.

IV. Extracting sub-assembly/part from assembly

You can extract the specified part or assembly by using extract target list.

Format of the extraction target file

An extract target list can be written by the XML format.

You can describe part or assembly names in the extract target list.

When the parameter "ExtractByRegularExpression" is equal to 1, the specified names and path names in the list are considered as a regular expression.

<AssemblyExtractionList> tag

You need to write the setting of this function between `AssemblyExtractionList` tags.
You can specify the assemblies/parts to extract by writing `AssemblyExtractionRule` tags between `AssemblyExtractionList` tags. You can write multiple Rule tags, too.

<AssemblyExtractionRule> tag

You need to write the setting of this function between `AssemblyExtractionRule` tags under `AssemblyExtractionList` tag.

You can specify the assemblies/parts to extract by writing `AssemblyExtractionRule` tags. Only one assembly/part will be extracted by one `AssemblyExtractionRule` tag. So you will need to write `AssemblyExtractionRule` tags as many as the number of assemblies you wish to extract.

You can specify the extraction manners by writing "type" attribute and/or "keep_position" attribute in `AssemblyExtractionRule` tags.

- **type** : (Mandatory)

Specify how to search for the object to extract. Here are 3 options of searching method.

- i. By part and/or assembly names (type="name")

Specify "Assembly" and/or "Part" tags.

Specified assembly or part will be extracted.

- ii. By path (type="namePath")

Specify "Part" tag nested in "Assembly" tags.

Specify the tree structure (path) from the top assembly to the target part by specifying the names of the assemblies and the part.

- iii. By instance path (type="instancePath")

Specify "Instance" tag nested in "Instance" tags.

Specify the tree structure (path) from the top assembly to the target instance by specifying the instance name.

- **keep_position** : (Option)

Specify "yes" or "no"

If keep_position = "yes", extract part with the parental assembly.

If keep_position = "no", extract part without the parental assembly.

"keep_position" attribute can be used only when type="name".

<Assembly> tag

Specify assembly name as extraction target.

Specify the following attribute in the Assembly tag.

name: (Mandatory)

Specify the name of the assembly to extract.

If ExtractByRegularExpression=1 is specified, names will be considered as strings using regular expression.

<Part> tag

Specify part name as extraction target.

Specify the following attribute in the Part tag.

name: (Mandatory)

Specify the name of the part to extract.

If ExtractByRegularExpression=1 is specified, names will be considered as strings using regular expression.

<Instance> tag

Specify instance name as extraction target.

Specify the following attribute in the Instance tag.

name: (Mandatory)

Specify the instance name of the assembly / part to extract.

If ExtractByRegularExpression=1 is specified, names will be considered as strings using regular expression.



- Tag and attribute names are case-sensitive.
- The order to describe the attribute is arbitrary.
- You must describe values according to the XML format. When you use the following characters in XML file, you must describe the string as follows.

```
'&':&amp;    '>':&gt;    '<':&lt;    '"':&quot;    '':&apos;
```

- Example of extract target list

```
<AssemblyExtractionList>

<!-- Rule 1 -->
  <AssemblyExtractionRule type="name" keep_position = "no">
    <Part name = "Part1" />
  </AssemblyExtractionRule>

<!-- Rule 2 -->
  <AssemblyExtractionRule type="name" keep_position = "yes">
    <Part name = "Part2"/>
  </AssemblyExtractionRule>

<!-- Rule 3 -->
  <AssemblyExtractionRule type="name" keep_position = "no">
    <Assembly name = "SubAsm2" />
  </AssemblyExtractionRule>
```

```

<!-- Rule 4 -->
  <AssemblyExtractionRule type="namePath">
    <Assembly name = "TopAsm">
      <Part name = "Part3" />
    </Assembly>
  </AssemblyExtractionRule>

<!-- Rule 5 -->
  <AssemblyExtractionRule type="namePath">
    <Assembly name = "TopAsm">
      <Assembly name = "SubAsm1">
        <Assembly name = "SubAsm2">
          <Part name = "Part1" />
        </Assembly>
      </Assembly>
    </Assembly>
  </AssemblyExtractionRule>

<!-- Rule 6 -->
  <AssemblyExtractionRule type="instancePath">
    <Instance name = "SubAsm1_instance1">
      <Instance name = "SubAsm2_instance1">
        <Instance name = "Part2_instance1" />
      </Instance>
    </Instance>
  </AssemblyExtractionRule>

</AssemblyExtractionList>

```

- Specification of the extract function

ENF is output as the result of extract. Target object name is used as output ENF name.

- If forbidden characters for Windows file is used, the characters are replaced by "_" (underscore).
- If type is namePath or instancePath, the ENF name becomes the name that is created by connecting each part, assembly and instance name with "_" (underscore).
- <If two or more target objects with the same name exist>
Each target object is extracted as a separate ENF. Each ENF name becomes the object name with serial number.
- <If target object is shared>
All objects are extracted as one ENF.
However, if keep_position=no, the shared objects are extracted as one object.

- Example of the extraction

Here is an example of extracting the assemblies below using the extracted list above.

- Original assembly structure.

- The string in parentheses is the instance name.
- SubAsm1 is shared.

```

TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|       |-----SubAsm2 (SubAsm2_instance1)
|               |-----Part1 (Part1_instance1)
|               |-----Part2 (Part2_instance1)
|
|-----SubAsm1 (SubAsm1_instance2)
|       |-----SubAsm2 (SubAsm2_instance1)
|               |-----Part1 (Part1_instance1)
|               |-----Part2 (Part2_instance1)
|
|-----Part3 (Part3_instance1)

```

1. Rule 1

Output file name: Part1.enf

An ENF file which includes only "Part1" will be exported. (Parent assemblies will not be included in that ENF file.)

2. Rule 2

Output file name: Part2.enf

An ENF file which includes "Part2" with parent assemblies will be exported.

The structure of the output ENF will be as follows:

```

TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|       |-----SubAsm2 (SubAsm2_instance1)
|               |-----Part2 (Part2_instance1)
|
|-----SubAsm1 (SubAsm1_instance2)
|       |-----SubAsm2 (SubAsm2_instance1)
|               |-----Part2 (Part2_instance1)
|

```

3. Rule 3

Output file name: SubAsm2.enf

An ENF file which includes "SubAsm2" with child parts will be exported.

The structure of the output ENF will be as follows:

```

SubAsm2
|-----Part1 (Part1_instance1)
|-----Part2 (Part2_instance1)

```

4. Rule 4

Output file name: TopAsm_Part3.enf

An ENF file which includes "Part3" with parent assembly will be exported.

The structure of the output ENF will be as follows:

```
TopAsm
|-----Part3 (Part3_instance1)
```

5. Rule 5

Output file name:

TopAsm_SubAsm1_SubAsm2_Part1_1.enf

TopAsm_SubAsm1_SubAsm2_Part1_2.enf

2 ENF files which includes "Part1" with parent assemblies will be exported.

The structure of the output ENF will be as follows:

```
TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part1 (Part1_instance1)
```

```
TopAsm
|-----SubAsm1 (SubAsm1_instance2)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part1 (Part1_instance1)
```

6. Rule 6

Output file name: SubAsm1_instance1_SubAsm2_instance1_Part2_insancet1.enf

An ENF file which includes "Part2" with parent assemblies will be exported.

The structure of the output ENF will be as follows:

```
TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part2 (Part2_instance1)
```

3.3.6. Interference Checker

By using this component, you can check interference of touch between parts in an assembly. By using CheckMode parameter of INTRCHECK section, you can specify check mode as interference or touch.

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, "XML Scenario"](#).

You can select "interference check" or "touch check" by the parameter.

```
[INTRCHECK]
CheckMode=1
```

0: interference check

1: touch check

(default value: 0)

II. Interference Checking

You can check interference between parts in an assembly. Output file is written in csv format. For detail, please refer parameter manual.

You can confirm the interference result on Model Viewer by importing the output csv file and the original ENF file to the Model Viewer. Please refer to the help file of Model Viewer for the detail.

III. Touch Checking

You can check touch or not between parts in an assembly. Output file is written in csv format. For detail, please refer parameter manual.

3.3.7. Polygon Optimizer

It optimizes specified polygon data. Following features are available.

- Automatic healing
- Smoothing Polygon
- Sharpen Edge
- Remesh
- Wrapping
- Offset
- Fill Hole
- Stitch
- Repair All Sharp Edge
- Delete Sharp Edge And Fill Hole
- Repair All Short Edge
- Repair All Long Edge
- Repair All Sliver Triangle
- Repair All Small Shell

- Repair All Small Volume
- Repair All Inner Voids
- Simplify Polygon

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, “XML Scenario”](#).

II. Set optimization categories

Add optimization items and process in the parameter file as shown below.

See the parameter manual for details.

- Example

Process_0=AutoHeal	1st step: Auto Heal
ModifyPolygonSmallShell_0=1	
ModifyPolygonSmallVolume_0=1	
ModifyPolygonCavity_0=1	
Process_1=Remesh	2nd step: Remesh
Process_2=Smooth	3rd step: Smoothing
ProcessNumberMax=3	Number of steps

III. Restriction

If data have wrong topology such as Singular Vertex, Complex Edge, Inconsistent Orientation and Duplicate Triangle, functions other than AutoHeal and Wrapping cannot be executed. In such cases, execute AutoHeal in advance.

The function Offset cannot be executed if a model has the vertex that its all surrounding triangle has extremely short edge or small inner angle. In such cases, heal faces that have short edge or small inner angle in advance.

3.3.8. ENF Editor

This is a component to edit ENF files. Below are two editing options.

- Writing any script
- Specifying a configuration file using pre-installed scripts
(This is to edit assemblies or attributes.)

3.3.8.1. Executing any Script

You can write any script using ENF Editor API for Ruby which allows you to obtain and edit the information of ENF files. This enables you to perform following operations.

- Adding, deleting and mapping user attributes/system attributes of components
- Calculating the number of components, volumes or face elements, and then exporting the result in a log file
- Editing the color, the status of show/hide or the layers of volumes, faces and edges
- Giving the layer number of the volume to its name
- Hiding volumes in specific layers

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, “XML Scenario”](#).

- XML scenario sample

```
<?xml version="1.0"?>
<ScenarioList>
  <Scenario>
    <ENFEDITOR>
      <scriptfile path="C:\in\user_script.rb"/>
      <inputfile path="C:\in\test.enf"/>
      <outputfile path="C:\out\test.enf"/>
      <logfile path="test.log"/>
      <xmllogfile path="test.xml"/>
      <workdir path="C:\work"/>
    </ENFEDITOR>
  </Scenario>
</ScenarioList>
```

II. How to write the scripts

Please refer to separate documents "ENF Editor Manual" and "ENF Editor API Reference Guide" for details.

3.3.8.2. Executing Pre-installed Scripts (Editing Assemblies)

You can edit assembly structures using the pre-installed scripts. Specify the configuration files, which is in Elysium-fixed format, in the script to perform the following operations. The configuration files are parameter files, namelists (the list of items to delete), "extraction_target" lists (the list of items to extract) and so on.

- [Deleting parts and assemblies](#)
- [Editing assembly structure](#)
- [Extracting parts and assemblies](#)

This script will make the same behavior as Assembly Editor up to EX 5.x. You can re-use namelists (the list of items to delete) and "extraction_target" lists (the list of items to extract).

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, “XML Scenario”](#).

- XML scenario sample (Deleting part/assembly)

```
<?xml version="1.0"?>
<ScenarioList>
  <Scenario>
    <ENFEDITOR>
      <scriptfile path="C:\ELYSIUM\ASFALIS_Components\module\tools\win\
elyEnfEditor\scripts\elysium\assembly_editor.rb"/>
      <inputfile path="C:\in\test.enf"/>
      <outputfile path="C:\out\test.enf"/>
      <namelist path="C:\in\deleteList.txt"/>
      <parameterfile path="C:\param\param.txt"/>
      <logfile path="test.log"/>
      <xmllogfile path="test.xml"/>
      <workdir path="C:\work"/>
    </ENFEDITOR>
  </Scenario>
</ScenarioList>
```

- XML scenario sample (Extracting sub-assembly/part from assembly)

```
<?xml version="1.0"?>
<ScenarioList>
  <Scenario>
    <ENFEDITOR>
      <scriptfile path="C:\ELYSIUM\ASFALIS_Components\module\tools\win\
elyEnfEditor\scripts\elysium\assembly_editor.rb"/>
      <inputfile path="C:\in\test.enf"/>
      <outputfile path="C:\out\test.enf"/>
      <extraction_target path="C:\in\extraction_target.xml"/>
      <extraction_outputdir path="C:\out"/>
      <parameterfile path="C:\param\param.txt"/>
      <logfile path="test.log"/>
      <xmllogfile path="test.xml"/>
      <workdir path="C:\work"/>
    </ENFEDITOR>
  </Scenario>
</ScenarioList>
```

II. Deleting part/assembly

You can delete part or assembly by using delete file list.

- Format of the namelist file.
 - The namelist file is a text file which lists the name of the parts and the assemblies to delete. The names are delimited by CR/LF code. If UseRegexpInNameList=1 is specified, the listed names will be considered as strings using regular expression.
 - If DeleteSpecifiedPart=1 is specified, a part will be deleted when its name exactly matches one of the names in the namelist. And assemblies, if DeleteSpecifiedAssembly=1 is specified, an assembly will be deleted when its name exactly matches one of the names in the namelist.
- Specification
 - When there exists more than one component under the same name which matches one of the names in the namelist, all of them will be deleted.
 - The parent assembly will remain (will not be deleted) even if it becomes empty after deleting the components underneath.
 - When deleting an assembly, the child parts and child assemblies underneath will also be deleted. In case those child parts or assemblies are shared by other assemblies, only that instance will be deleted and the shared parts or assemblies at other instances will remain.
 - If the namelist file is not specified, does not exist or cannot be read, the process afterward will not be executed, and the program will be closed. It will be written as "Failed to open list" in the log file.
 - If the component to delete which is listed in the namelist does not exist, it will be skipped and move onto the next one. It will be written as "Component XXX was not found in the model" in the log file.

III. Editing Assembly

You can edit assembly structure by using this component. For detail, please refer parameter manual.

IV. Extracting sub-assembly/part from assembly

You can extract the specified part or assembly by using extract target list.

- Format of the extraction target file

An extract target list can be written by the XML format.
You can describe part or assembly names in the extract target list.
When the parameter "ExtractByRegularExpression" is equal to 1, the specified names and path names in the list are considered as a regular expression.

<AssemblyExtractionList> tag

You need to write the setting of this function between AssemblyExtractionList tags.
You can specify the assemblies / parts to extract by writing AssemblyExtractionRule tags between AssemblyExtractionList tags. You can write multiple Rule tags, too.

<AssemblyExtractionRule> tag

You need to write the setting of this function between AssemblyExtractionRule tags under AssemblyExtractionList tag.

You can specify the assemblies / parts to extract by writing AssemblyExtractionRule tags. Only one assembly / part will be extracted by one AssemblyExtractionRule tag. So you will need to write AssemblyExtractionRule tags as many as the number of assemblies you wish to extract.

You can specify the extraction manners by writing "type" attribute and/or "keep_position" attribute in AssemblyExtractionRule tags.

- **type:** (Mandatory)

Specify how to search for the object to extract. Here are 3 options of searching method.

- i. By part and/or assembly names (type="name")

Specify "Assembly" and/or "Part" tags.

Specified assembly or part will be extracted.

- ii. By path (type="namePath")

Specify "Part" tag nested in "Assembly" tags.

Specify the tree structure (path) from the top assembly to the target part by specifying the names of the assemblies and the part.

- iii. By instance path (type="instancePath")

Specify "Instance" tag nested in "Instance" tags.

Specify the tree structure (path) from the top assembly to the target instance by specifying the instance name.

- **keep_position:** (Option)

Specify "yes" or "no"

If keep_position = "yes", extract part with the parental assembly.

If keep_position = "no", extract part without the parental assembly.

"keep_position" attribute can be used only when type="name".

<Assembly> tag

Specify assembly name as extraction target.

Specify the following attribute in the Assembly tag.

name: (Mandatory)

Specify the name of the assembly to extract.

If ExtractByRegularExpression=1 is specified, names will be considered as strings using regular expression.

<Part> tag

Specify part name as extraction target.

Specify the following attribute in the Part tag.

name: (Mandatory)

Specify the name of the part to extract.

If ExtractByRegularExpression=1 is specified, names will be considered as strings using regular expression.

<Instance> tag

Specify instance name as extraction target.

Specify the following attribute in the Instance tag.

name: (Mandatory)

Specify the instance name of the assembly / part to extract.

If ExtractByRegularExpression=1 is specified, names will be considered as strings using regular expression.



- Tag and attribute names are case-sensitive.
- The order to describe the attribute is arbitrary.
- You must describe values according to the XML format. When you use the following characters in XML file, you must describe the string as follows.

'&':& '>':> '<':< '"':" '':'

- Example of extract target list

```

<AssemblyExtractionList>

<!-- Rule 1 -->
  <AssemblyExtractionRule type="name" keep_position = "no">
    <Part name = "Part1" />
  </AssemblyExtractionRule>

<!-- Rule 2 -->
  <AssemblyExtractionRule type="name" keep_position = "yes">
    <Part name = "Part2"/>
  </AssemblyExtractionRule>

<!-- Rule 3 -->
  <AssemblyExtractionRule type="name" keep_position = "no">
    <Assembly name = "SubAsm2" />
  </AssemblyExtractionRule>

<!-- Rule 4 -->
  <AssemblyExtractionRule type="namePath">
    <Assembly name = "TopAsm">
      <Part name = "Part3" />
    </Assembly>
  </AssemblyExtractionRule>

<!-- Rule 5 -->
  <AssemblyExtractionRule type="namePath">
    <Assembly name = "TopAsm">
      <Assembly name = "SubAsm1">
        <Assembly name = "SubAsm2">
          <Part name = "Part1" />
        </Assembly>
      </Assembly>
    </Assembly>
  </AssemblyExtractionRule>

<!-- Rule 6 -->
  <AssemblyExtractionRule type="instancePath">
    <Instance name = "SubAsm1_instance1">
      <Instance name = "SubAsm2_instance1">
        <Instance name = "Part2_instance1" />
      </Instance>
    </Instance>
  </AssemblyExtractionRule>

</AssemblyExtractionList>

```

- Specification of the extract function

ENF is output as the result of extract. Target object name is used as output ENF name.

- If forbidden characters for WINDOWS file is used, the characters are replaced by "_" (underscore).
 - If type is namePath or instancePath, the ENF name becomes the name that is created by connecting each part, assembly and instance name with "_" (underscore).
 - <If two or more target objects with the same name exist>
Each target object is extracted as a separate ENF. Each ENF name becomes the object name with serial number.
 - <If target object is shared>
All objects are extracted as one ENF.
However, if keep_position=no, the shared objects are extracted as one object.
- Example of the extraction

Here is an example of extracting the assemblies below using the extracted list above.

- Original assembly structure.
- The string in parentheses is the instance name.
- SubAsm1 is shared.

```

TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part1 (Part1_instance1)
|                   |-----Part2 (Part2_instance1)
|
|-----SubAsm1 (SubAsm1_instance2)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part1 (Part1_instance1)
|                   |-----Part2 (Part2_instance1)
|
|-----Part3 (Part3_instance1)

```

1. Rule 1

Output file name: Part1.enf

An ENF file which includes only "Part1" will be exported. (Parent assemblies will not be included in that ENF file.)

2. Rule 2

Output file name: Part2.enf

An ENF file which includes "Part2" with parent assemblies will be exported.

The structure of the output ENF will be as follows:

```

TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part2 (Part2_instance1)
|
|-----SubAsm1 (SubAsm1_instance2)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part2 (Part2_instance1)

```

3. Rule 3

Output file name: SubAsm2.enf

An ENF file which includes "SubAsm2" with child parts will be exported.

The structure of the output ENF will be as follows:

```

SubAsm2
|-----Part1 (Part1_instance1)
|-----Part2 (Part2_instance1)

```

4. Rule 4

Output file name: TopAsm_Part3.enf

An ENF file which includes "Part3" with parent assembly will be exported.

The structure of the output ENF will be as follows:

```

TopAsm
|-----Part3 (Part3_instance1)

```

5. Rule 5

Output file name:

TopAsm_SubAsm1_SubAsm2_Part1_1.enf

TopAsm_SubAsm1_SubAsm2_Part1_2.enf

2 ENF files which includes "Part1" with parent assemblies will be exported.

The structure of the output ENF will be as follows:

```

TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part1 (Part1_instance1)

```

```

TopAsm
|-----SubAsm1 (SubAsm1_instance2)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part1 (Part1_instance1)

```

6. Rule 6

Output file name: SubAsm1_instance1_SubAsm2_instance1_Part2_insancet1.enf

An ENF file which includes "Part2" with parent assemblies will be exported.

The structure of the output ENF will be as follows:

```
TopAsm
|-----SubAsm1 (SubAsm1_instance1)
|           |-----SubAsm2 (SubAsm2_instance1)
|                   |-----Part2 (Part2_instance1)
```

3.3.8.3. Executing Pre-installed Scripts (Editing Attributes)

You can edit attributes using the pre-installed scripts. Specify mapfiles and bomfiles etc. in the script.

- [Mapping attributes](#)
- [Extracting attributes](#)
- [Replacing attribute values](#)
- [Adding attribute values](#)
- [Editing instance names](#)

This script will make the same behavior as Attribute Editor up to EX 5.x. You can re-use mapfiles.

I. How to use

This function can be used by defining the XML scenario. Please refer to [2.4, "XML Scenario"](#).

```
<?xml version="1.0"?>
<ScenarioList>
  <Scenario>
    <ENFEDITOR>
      <scriptfile path="C:\ELYSIUM\ASFALIS_Components\module\tools\win\
elyEnfEditor\scripts\elysium\attribute_editor.rb"/>
      <inputfile path="C:\in\test.enf"/>
      <outputfile path="C:\out\test.enf"/>
      <mapfile path="C:\in\mapping.xml" />
      <logfile path="test.log"/>
      <xmllogfile path="test.xml"/>
      <workdir path="C:\work"/>
    </ENFEDITOR>
  </Scenario>
</ScenarioList>
```

II. Editing attribute

The following operations are possible. The following operation is individually described respectively as "Rule". It is possible to specify two or more "Rule" at the same time.

1. Create (type = "create")

Convert the value of attribute from "FROM" to "TO".

- Format

Attribute key and type of "From" and "To" are defined by one rule.

Type is specified by either character string of "System" or "User".

2. Delete (type = "delete")

Delete the specified attributes.

- Format

Enumerate the deleted attribute.

3. Combine (type = "combine")

Connect the specified attributes and with the character string, and create a new attribute.

- Format

Arrange attributes that unites and character string in "From" tag.

You can select items from attribute, string and Serial Number.

"Serial Number" is a number that is incremented by 1 as attributes get created. (Initial value is 0.)

- Attribute: Target attribute
- String: String which is combined with value attribute
- Serial Number: Specify digit number by using "digit" attribute

By specifying the "target" attribute for each Rule, you can specify the object element of the operation.

- Sample


```
<?xml version="1.0" encoding="Shift_JIS"?>
<AttrEdit>
  <MappingRules>
    <Rule type="create">
      <From key="Author" type="user"/>
      <To key="Creator" type="user"/>
    </Rule>
    <Rule type="delete">
      <Attribute key="Title" type="user"/>
      <Attribute key="Author" type="user"/>
    </Rule>
    <Rule type="combine" target="assembly">
      <From>
        <Attribute key="Name" type="system"/>
        <String value="_"/>
        <Attribute key="PartNumber" type="system"/>
      </From>
      <To key="Name" type="system"/>
    </Rule>
    <Rule type="combine" target="part">
      <From>
        <Attribute key="Name" type="system"/>
        <String value="_"/>
        <SerialNumber digit="4"/>
      </From>
      <To key="Name" type="system"/>
    </Rule>
  </MappingRules>
</AttrEdit>
```

- The Rule will be ignored when the format is incorrect in the Rule tag.
- Only the first rule will be used and all the rest will be ignored when there exists more than one rule in which the type of the target attribute duplicates each other. Warning message will be written in the log file, too
- You must describe values according to the XML format. When you use the following characters in XML file, you must describe the string as follows.

```
'&':&amp;    '>':&gt;    '<':&lt;    '"':&quot;    '':&apos;
```

e.g.,)

not available: <String value="Name&Number"/>

available: <String value="Name&Number"/>



- When you use "force" attributes and specify "on" for that in the "Rule" tag, attributes will be edited even if the value of the target attribute is empty. e.g.,) <Rule type="combine" force="on">
- Only the attributes with the following keys will be regarded as "System" properties.

Name	PartName	PartNumber
NFName	ConfigName	BinName
ChangeHistory	Description	PartComment
PartDefinition	PartVersion	PartRevision
PartSource	TFName	PartLayer

III. Extracting attribute

Attribute information included in neutral file can be written by the XML form as another file. The output file is specified for "bomfile" in the XML scenario.

- Sample of the BOM file.

```

<?xml version="1.0" encoding="UTF-8"?>
<AttrList>
  <Owner name="bom_sample_out" type="model">
  </Owner>
  <Owner name="top_asm_sample" type="assembly">
    <Attr class="system" key="PartNumber" type="text">sample</Attr>
  </Owner>
  <Owner name="Asm1_01" type="assembly">
    <Attr class="system" key="PartNumber" type="text">01</Attr>
    <Attr class="system" key="PartRevision" type="text">12345</Attr>
  </Owner>
  <Owner name="part1" type="part">
    <Attr class="system" key="PartName" type="text">part1</Attr>
    <Attr class="system" key="PartNumber" type="text">part1_number</Attr>
  </Owner>
  <Owner name="Asm2_02" type="assembly">
    <Attr class="system" key="PartNumber" type="text">02</Attr>
    <Attr class="system" key="PartRevision" type="text">67890</Attr>
  </Owner>
  <Owner name="part2" type="part">
    <Attr class="system" key="PartName" type="text">Part2</Attr>
  </Owner>
  <Owner name="part3" type="part">
    <Attr class="system" key="Description" type="text">UserAttribute</Attr>
    <Attr class="system" key="Material" type="text">Default</Attr>
    <Attr class="system" key="PartName" type="text">part3</Attr>
    <Attr class="user" key="Creator" type="text">yamada</Attr>
    <Attr class="user" key="Subject" type="text">test_model</Attr>
  </Owner>
</AttrList>

```

IV. Replacing attribute value

You can replace attribute value by using ValueReplacementRules.

<ValueReplacementRules> tag

You need to write the setting of this function between ValueReplacementRules tags in AttrEdit tag. You can write multiple Rule tags, too.

<Rule> tag

Following options can be set as the attribute. (Option)

target: Specify the element type from "part," "assembly" or "model." Only the attributes of specified element types will be replaced. All attributes will be replaced in case you don't specify the target attributes.

regex: Specify "on" and also the "search and replace" values so that "search and/or replace" strings become regular expressions. Specify "off" or write nothing, then "search and replace"

value will not be regular expressions. (The exact value specified at "search" will be replaced to the value at "replace.")

Specify the target attribute (the key and the type) and the character string to replace from and to between the "Rule" tag.

Only the first rule will be used and all the rest will be ignored when there exists more than one rule in which the type of the target attribute duplicates each other. Warning message will be written in the log file, too.

<Attribute> tag

Specify target attribute by using the following attributes. (Mandatory)

key: Specify the attribute key.

type: Specify the attribute type, system property or user property by "system" or "user."

<Value> tag

Specify the value of the attribute to replace from and to. (Mandatory)

from: Specify the search string.

to: Specify the replacement string.

Write "Value" tag right after "Attribute" tag. You can specify multiple tags, too. All the strings which match the specified value will be replaced. Replacement will be performed from the first rule checking through all the attributes before moving on to the next rule. Once a string has been replaced, it will not be replaced again even when the string matches the search string in the tag(s) written below.

- Example

```
<?xml version="1.0"?>
<AttrEdit>
  <ValueReplacementRules>
    <Rule target="part" regexp="off">
      <Attribute key="Material" type="system"/>
      <Value from="Steel" to="Metal"/>
      <Value from="Copper" to="Bronze"/>
    </Rule>
    <Rule regexp="on">
      <Attribute key="Name" type="system"/>
      <Value from="^AAA.*" to="ABC"/>
    </Rule>
  </ValueReplacementRules>
</AttrEdit>
```

With the mapping rule above, regarding the "Material" property which is a system property, the value will be replaced from "Steel" to "Metal" and "Copper" to "Bronze."

And regarding the "Name" property which is also a system property, all the values beginning with "AAA" will be replaced to "ABC".

V. Adding attribute value

You can set attribute value by using ValueAdditionRules.

<ValueAdditionRules> tag

You need to write the setting of this function between ValueAdditionRules tags in AttrEdit tag. You can write multiple Rule tags, too.

Following options can be set as the attribute. (Option)

overwrite: Specify "on" or "off." When "on" is specified, the value will be overwritten even if an attribute under the same name already exists. It will be regarded as "off" when the attribute "overwrite" is not specified. A system property and a user property will be regarded as the different attributes even if they have the same attribute key.

<Rule> tag

Following options can be set as the attribute. (Option)

target: Specify the element type from "part," "assembly" or "model." Attributes will be added only to the attributes of specified element types.

You need to write the name of the attribute to add attributes to, and the attribute value to add between "Rule" tags.

<TargetName> tag

Specify the name of the target part or assembly as the value attribute of the tag. The attribute will be added only when the name exactly matches the specified value. All elements will be the target when this tag is not specified. The attribute will be added to the model regardless whether "TargetName" tag is specified or what is specified for "value" attribute, when the target="model" is specified in the Rule tag.

Following options can be set as the attribute.

regexp: Specify "on" and also the "search" value so that "search" string becomes a regular expression. Specify "off" or write nothing, then "search" value will not be regular expressions.

<Attribute> tag

Specify target attribute by using the following attributes. (Mandatory)

key: Specify the attribute key.

type: Specify the attribute type, system property or user property by "system" or "user."

<Value> tag

Specify the value to be added of the attribute. You need to write the attribute value to add between "Value" tags. (Mandatory)

- Example

```
<?xml version="1.0"?>
<AttrEdit>
  <ValueAdditionRules overwrite="on">
    <Rule target="part">
      <TargetName regexp="on" value="^Arm200-101-\d*$"/>
      <Attribute key="Material" type="system"/>
      <Value>copper</Value>
    </Rule>
    <Rule target="assembly">
      <TargetName regexp="on" value="^Arm200-asm-\d*$"/>
      <Attribute key="Maker" type="user"/>
      <Value>ABC Corp.</Value>
    </Rule>
    <Rule target="model">
      <Attribute key="Designer" type="user"/>
      <Value> Yamada Taro</Value>
    </Rule>
  </ValueAdditionRules>
</AttrEdit>
```

With the mapping rule above, the attribute "Material" will be added to the parts whose name begin with "Arm200-101," and the attribute "Maker" will be added to the assemblies whose name begin with "Arm200-asm-."

Also the attribute "Designer" will be added to the model.

VI. Editing instance name

You can edit instance name by using TargetSpecificRules.

<TargetSpecificRules> tag

You need to write the setting of this function between TargetSpecificRules tags in AttrEdit tag.

<Rule> tag

Specify following options as the attribute.

type="combine"

target="instance"

target: "instance" is effective only when editing instance name.

<From> tag

Specify the items you want to connect in the "From" node.

You can select items from attribute, string and Serial Number.

<String> tag

value: Specify the character string you want to use for connecting.

<SerialNumber> tag

digit: Specify digit number (selectable range: 1-20, omissible).

<Attribute> tag

Specify following options as the attribute.

source: "Parent" or "Child" source

"Parent" means the upper assembly of the instance.

"Child" means the child part or assembly of the instance.

key: Specify the attribute key of the element that specified by "source".

type: Specify the "system" attribute or "user" attribute.

<To> tag

Specify following option as the attribute.

key="Name"

- Example

You can get the instance name (with sequence number) by combining parent assembly name and child part/assembly name via "_" (underscore).

```
<AttrEdit>
  <TargetSpecificRules>
    <Rule type="combine" target="instance">
      <From>
        <Attribute source="Parent" key="Name" type="system"/>
        <String value="_"/>
        <Attribute source="Child" key="Name" type="system"/>
        <SerialNumber digit="1"/>
      </From>
      <To key="Name"/>
    </Rule>
  </TargetSpecificRules>
</AttrEdit>
```

3.3.9. 3D PDF Editor

This is a Component to customize the content to include in the output 3D PDF file, layout design of the output 3D PDF file etc.

I. How to use 3D PDF Editor

This function can be used by defining the XML scenario. Please refer to [2.4, "XML Scenario"](#).

II. Restriction

When customizing 3D PDF report, use ASFALIS DATA Package Studio. Please refer to the document, "ASFALIS DATA Package Studio" for more details.

3.4. Other

3.4.1. 3D Annotation Translation

Please refer to "Requirement_ENG.pdf" for CAD formats and versions which 3D annotation translation is supported.

I. License

It requires an additional license on 3D annotation translation (ASF-ANN/<Code of target CAD>) to translate including 3D annotations in ENF to CAD translation.

Please refer to "Requirement_ENG.pdf" for the product code of necessary licenses.



The license for ENF Writer (CAD Import) Component includes the translation of 3D annotations and attributes as well.

II. Parameters to Control Settings on 3D Annotation Translation

Set following parameters as required.

- Parameter Key Description

Parameter Key	Description
XConvertNote (*1)	Specify whether to convert Note.
XConvertGDT (*1)	Specify whether to convert GDT, Datum, and Datum Target.
XConvertDimension (*1)	Specify whether to convert Dimension.
XConvertSurfFinish (*1)	Specify whether to convert Surface Finish Symbol.
XConvertSpotWeld (*1) XConvertLineWeld (*1)	Specify whether to convert Welding Symbol.
XConvertLocator (*1)	Specify whether to convert Locator.
XConvertView	Specify whether to convert Capture and View.
XConvertSection	Specify whether to convert Section.
XConvertHatching	Specify whether to convert Hatching.
XConvert3DAnnotationAsPolyline	Specify whether to add polyline for PMI as well as converting as semantic.
XConvertSymbol	Specify whether to convert User Defined Symbol.
XConvertGraphicDataPMI	Specify whether to convert graphic PMI such as center line and region as graphic PMI.

(*1): In ENF to CAD translation, you can also choose to translate this element as polyline (symbols / isolated lines) to maintain the graphical representation.
(Available for specific CAD formats only)

III. Supported PMI Elements

1. CAD to ENF

- CATIA V5 to ENF

Category	CATIA V5 FTA
Note	Text / Flag Note
Datum	Datum Feature
Datum Target	Datum Target
Geometrical tolerance	Geometrical Tolerance
Surface finish	Roughness
Dimension	Dimension
Welding	SpotPoint Weld Symbol

- 3DEXPERIENCE to ENF

Category	3D Tolerancing & Annotation
Note	Text / Flag Note
Datum	Datum Feature
Datum Target	Datum Target
Geometrical tolerance	Geometrical Tolerance
Surface finish	Roughness
Dimension	Dimension
Welding	Weld Feature



All the items are translated as polyline.

- NX I-deas to ENF

Category	NX I-deas PMI
Note	Note
Datum	Datum Feature Symbol
Datum Target	Datum Target
Geometrical tolerance	Feature Control Frame
Surface finish	Surface Finish
Dimension	Dimension
Welding	SpotWeld LineWeld
Locater	Locator

- Creo Parametric to ENF

Category	Creo Parametric
Note	Note
Symbol	Symbol
Geometrical tolerance	Geometric Tolerance
Surface finish	Surface Finish
Dimension (*1)	Dimension
Set datum tag	Set Datum Tag
Non-graphical	Non-Graphical

- (*1): You can control whether to display the tolerances of dimensions by the following parameter either in the configuration settings of Creo Parametric, or in the configuration file for Creo Parametric adapter located in the following file path.

tol_display=no	Display dimensions without tolerances
tol_display=yes	Display dimensions with tolerances



- Please note that the parameter set in "config.sup" of Creo Parametric used for translation will be prioritized when set in both.
- Please note that this parameter is commented out by default in both configuration files.

```
!tol_display=yes
```

File location of Creo Parametric Adapter configuration file

- Legacy Adapter
 <ASFALIS Components>\ProeExe\win\<Creo Version>\config.pro
- New Adapter
 <ASFALIS Components>
 \CADFeature\Batch\Creo.x64\Parametric\<Creo Version>\config.pro

◦ NX to ENF

Category	NX
Note	Note Drafting Note Specialized Note
Datum	Datum Feature Symbol Drafting Datum Feature Symbol Geometrical Tolerancing-Datum
Datum Target	Datum Target Geometrical Tolerancing-Datum Target
Geometrical tolerance	Feature Control Frame Drafting Feature Control Frame Geometrical Tolerancing-Tolerance
Surface finish	Surface Finish
Dimension	Dimension Drafting Dimension Geometrical Tolerancing-Dimension
Welding	SpotWeld LineWeld
Custom symbol	Custom Symbol
Locator	Locator

Category	NX
Center line	Center Line Center Mark
Region	Region

◦ SOLIDWORKS to ENF

Category	SOLIDWORKS
Note	Note Balloon
Datum	Datum Feature
Datum Target	Datum Target
Geometrical tolerance	Geometric Tolerance
Dimension	Dimension DimXpert
Surface finish	Surface Finish
Welding	Weld Symbol



It is required to use a new adapter to translate PMI. You cannot translate PMI on the legacy adapter.

◦ JT to ENF

Category	JT
Note	Note Coordinate Note Balloon Specialized Note
Datum	Datum Feature Symbol
Datum Target	Datum Target
Geometrical tolerance	Feature Control Frame
Surface finish	Surface Finish
Dimension	Dimension

Category	JT
Welding	SpotWeld LineWeld
User defined symbol	User Defined Symbol
Center line	Center Line Center Mark
Region	Region
Locator	Locator
Cutting plane symbol	Cutting Plane Symbol

- STEP to ENF

Category	STEP
Note	Note Label General Note
Datum	Datum
Datum Target	Datum Target
Geometrical tolerance	Geometric Tolerance
Surface finish	Surface Roughness
Dimension	Dimension
Welding	Weld Symbol

- CADmeister to ENF

Category	CADmeister
Note	3D Note
Dimension	Ref. Dimension

2. ENF to CAD

- ENF to CATIA V5

Category	CATIA V5 FTA
Note	Text / Flag Note

Category	CATIA V5 FTA
Datum	Datum Feature
Datum Target	Datum Target
Geometrical tolerance	Geometrical Tolerance
Surface finish	Roughness
Dimension	Dimension
Welding	SpotPoint (*1) Weld Symbol
Locater	Converted as polyline

- (*1): This will be translated as isolated point.



Please note that it requires an additional license on 3D annotation translation ("ENF to CATIA V5 3D Annotation" option) to translate 3D annotations to CATIA V5.

- ENF to 3DEXPERIENCE

Category	3D Tolerancing & Annotation
Note	Text / Flag Note
Datum	Datum Feature
Datum Target	Datum Feature
Geometrical tolerance	Geometrical Tolerance
Surface finish	Roughness
Dimension	Dimension
Welding	Weld Feature
Locater	Text



All the items are translated as polyline.

- ENF to NX I-deas

Category	NX I-deas PMI
Note	Note

Category	NX I-deas PMI
Datum	Datum Feature Symbol
Datum Target	Datum Target
Geometrical tolerance	Feature Control Frame
Surface finish	Surface Finish
Dimension	Dimension
Welding	SpotWeld LineWeld
Locator	Locator



Please note that it requires;

- An additional license on 3D annotation translation ("ENF to I-deas 3D Annotation" option)
- NX I-deas "geometric_tol" license

to translate 3D annotations to NX I-deas.

◦ ENF to NX

Category	NX
Note	Note
Datum	Datum Feature Symbol
Datum Target	Datum Target
Geometrical tolerance	Feature Control Frame
Surface finish	Surface Finish
Dimension	Dimension
Custom symbol	Custom Symbol



Please note that it requires;

- An additional license on 3D annotation translation ("ENF to NX 3D Annotation" option)
- NX "drafting" and "geometric_tol" licenses

to translate 3D annotations to NX.

Please note that some element types will not be translated.

◦ ENF to JT

Category	JT
Note	Note Coordinate Note
Datum	Datum Feature Symbol
Datum Target	DatumTarget
Geometrical tolerance	Feature Control Frame
Surface finish	Surface Finish
Dimension	Dimension
Welding	SpotWeld LineWeld
Locator	Locator
User defined symbol	User Defined Symbol



Please note that it requires an additional license on 3D annotation translation ("ENF to JT 3D Annotation" option) to translate 3D annotations to JT.

◦ ENF to STEP

Category	STEP
Note	Note
Datum	Datum
Datum Target	Datum Target
Geometrical tolerance	Geometric Tolerance

Category	STEP
Surface finish	Surface Roughness
Dimension	Dimension
Welding	Weld Symbol



- Please note that it requires an additional license on 3D annotation translation ("ENF to STEP 3D Annotation" option) to translate 3D annotations to STEP.

- ENF to CADmeister

Category	CADmeister
Note	3D Note
Polyline Dimension	Translate as a line element according to the appearance of polyline

- ENF to 3D PDF

Category	3D PDF
Note	Text
Datum	Datum
Datum Target	Datum
Geometrical tolerance	Gdt
Surface finish	Roughness
Dimension	Dimension
Locator	Locator

IV. Capture / View

1. CAD to ENF

- CATIA V5 to ENF

Element	Description
Capture	<ul style="list-style-type: none"> - Translates Capture that belongs to parts or assemblies. - Translates viewpoints, display attributes (show/hide status) of annotations (FTA / welding symbol), and one plane sections (clipping plane) that belong to Capture.
Named View	<ul style="list-style-type: none"> - Translates the view directions saved as Named View that belong to parts or assemblies.

- 3DEXPERIENCE to ENF

Element	Description
Capture	<ul style="list-style-type: none"> - Translates Capture that belongs to parts. - Translates viewpoints, display attributes (show/hide status) of annotations, and one plane sections (clipping plane) that belong to Capture.
View (Camera)	<ul style="list-style-type: none"> - Translates the view directions saved as Named Views that belong to parts.

- NX I-deas to ENF

Element	Description
Model View	<ul style="list-style-type: none"> - Translates Model View that belongs to parts or assemblies.



Model View

- Does not support sections that belong to Model View.
- Does not support display attributes (show/hide status) of PMI that belong to Model View.

- NX to ENF

Element	Description
Model View	<ul style="list-style-type: none"> - Translates Model View that belongs to parts or assemblies. - Translates display attributes (show/hide status) of 3D PMI and display attributes of layers that belong to Model View. - Translates sections (the sections of PMI section view and View Section) that belong to Model View.

- Creo Parametric to ENF

Element	Description
Combined View	<ul style="list-style-type: none"> - Translates Combined View that belongs to parts or assemblies. - Translates viewpoints, sections, and display attributes (show/hide status) of annotations that belong to Combined View.
Orientation	<ul style="list-style-type: none"> - Translates the view directions saved as Orientation that belong to top parts or top assemblies.



Combined View - Section

- Supports single-plane sections and Zone Cross sections created in Creo Parametric 3.0 or later only.
- Does not support hatching on sections.

◦ SOLIDWORKS to ENF

Element	Description
Annotation View	<ul style="list-style-type: none"> - Translate Annotation View that belongs to top parts or top assemblies. - Translate viewpoints and display attributes (show/hide status) of annotations that belong to Annotation View.
3D View	<ul style="list-style-type: none"> - Translate 3D View that belongs to top parts or top assemblies. - Translate viewpoints, sections, and display attributes (show/hide status) of annotations that belong to 3D View.

◦ JT to ENF

Element	Description
Model View	<ul style="list-style-type: none"> - Translates Model View that belongs to parts and assemblies. - Translates viewpoints, display attributes (show/hide status) of annotations, display attributes of elements (parts, assemblies, and geometry elements), and sections that belong to Model View.

◦ STEP to ENF

Element	Description
Saved View	<ul style="list-style-type: none"> - Translates Saved View that belongs to parts. - Translates display attributes (show/hide status) of annotations that belong to Saved View. - Translates one plane sections that belong to parts.

2. ENF to CAD

◦ ENF to CATIA V5

Element	Description
Capture	<ul style="list-style-type: none"> - Translates Capture that belongs to parts or assemblies. - Translates viewpoints, display attributes (show/hide status) of annotations (FTA) that belong to Capture. - Translates sections that belong to parts or assemblies as one clipping plane.

◦ ENF to 3DEXPERIENCE

Element	Description
Capture	<ul style="list-style-type: none"> - Translates Capture that belongs to parts. - Translates viewpoints, display attributes (show/hide status) of annotations, and one plane sections(clipping plane) that belong to Capture.

◦ ENF to NX

Element	Description
Model View	<ul style="list-style-type: none"> - Translates Camera that belongs to parts or assemblies. - Translates display attributes (show/hide status) of annotations that belong to Camera. - Translates sections that belong to parts or assemblies as PMI section view.

**Model View**

Does not support display attributes (show/hide status) of part annotations included in Camera that belong to assemblies

◦ ENF to JT

Element	Description
Model View	<ul style="list-style-type: none"> - Translates Camera that belongs to parts or assemblies.
Section	<ul style="list-style-type: none"> - Translates sections that belong to parts or assemblies. - Translates display attributes (show/hide status) of annotations that belong to Camera. - Translates display attributes (show/hide status) of elements (parts, assemblies, and geometry elements) that belong to Camera.

◦ ENF to STEP

Element	Description
Saved View	<ul style="list-style-type: none"> - Translates Camera that belongs to parts. - Translates one plane sections that belong to parts.

V. Section

1. CAD to ENF

- NX I-deas to ENF

Element	Description
Section	- Translates the section names, outlines of the section plane, and orientations of the section as polyline

VI. Hatching

1. CAD to ENF

- NX I-deas to ENF

Element	Description
Hatching	- Translates hatchings as polyline



Hatching

Does not support hatching on sections.

- NX to ENF

Element	Description
Hatching	- Translates hatchings



Hatching

- Supports hatching on planes only.
- Hatching lines will be translated as solid lines regardless of the line type.

3.4.2. Attribute Translation

I. License

It requires an additional license on attribute translation (ASF-ATR/<Code of target CAD>) to

translate including attributes in ENF to CAD translation. Please refer to "Requirement_ENG.pdf" for the product code ("feature" in Sentinel RMS License Manager) of necessary licenses.

II. Parameters to Control Settings on Attribute Translation

Set following parameters as required.

Parameter Key	Description
XConvertSystemProperty	Specify whether to convert system attributes.
XConvertUserProperty	Specify whether to convert user attributes.
XConvertMaterial	Specify whether to convert material attributes.

III. Supported Attribute Elements

Terminology)

- System Attribute: Properties that are defined by CAD systems, CAD format etc. Attribute keys are unamendable.
- User Attribute: Properties that are defined by users in the format of "Key=Value".

1. CAD to ENF

- CATIA V5 to ENF

Type	Attribute in CATIA V5	Key Name in ENF
System Attribute	Nomenclature	PartName
	PartNumber	PartNumber
	Revision	PartRevision
	Definition	PartDefinition
	Description	Description
	Source	PartSource
	Part Layer	PartLayer
	Description(Instance)	InstanceDescription
	Material Name	Material
User Attribute	Other Property	(User definition)



- Supports attributes that belong to following elements only:
 - CATProduct
 - CATPart
 - Shell (Supports material attributes only)
 - Lump (Supports material attributes only)
 - Geometry set (Supports User Attribute only)
 - User defined feature (Supports User Attribute only)

In CATIA V5 to NX I-deas translation, translates while mapping as follows in Auto Healing process:

- "PartNumber" in CATIA V5 to "Part Number" in NX I-deas
- "Nomenclature" in CATIA V5 to "Name" in NX I-deas

◦ 3DEXPERIENCE to ENF

Type	Attribute in 3DEXPERIENCE	Key Name in ENF
System Attribute	Name	PartName
	Description	Description
	Revision	PartRevision
	Last modification	NativeFileLastWriteTime
User Attribute	Attributes	(User definition)



- Supports attributes that belong to following elements only:
 - Product
 - 3D Part

◦ NX I-deas to ENF

Type	Attribute in NX I-deas	Key Name in ENF
System Attribute	Name	PartName
	Part Number	PartNumber
	Configuration Name	ConfigName
	Bin Name	BinName
	Version	PartVersion
	Revision	PartRevision
	Description	Description
	Change History	ChangeHistory
	Material Name	Material
User Attribute	User Property	(User definition)



- Supports attributes that belong to following elements only:
 - Assembly
 - Part

In NX I-deas to CATIA V5 translation, translates while mapping as follows in Auto Healing process:

- "Part Number" in NX I-deas to "PartNumber" in CATIA V5
- "Name" in NX I-deas to "Nomenclature" in CATIA V5

◦ NX to ENF

Type	Attribute in NX	Key Name in ENF
System Attribute	Material	Material
	input file last write time	NativeFileLastWriteTime
	input file size	NativeFileSize
User Attribute	Property	(User definition)



- Supports attributes that belong to following elements only:
 - Part (*1)
 - Body (Supports SolidBody and SheetBody only)
 - Face (*1)
 - Edge (*1)
 - Curve (*1)
 - Point (*1)
- (*1): Does not support material attributes that belong to this element.

◦ Creo Parametric to ENF

Type	Attribute in Creo Parametric	Key Name in ENF
System Attribute	Revision	PartRevision
	Material	Material
User Attribute	Parameter	(User definition)



- Supports attributes that belong to following elements only:
 - Assembly
 - Part
 - Datum Point
 - Edge
 - Surface
 - Annotation Feature

◦ Inventor to ENF

Type	Attribute in Inventor	Key Name in ENF
System Attribute	Comments	Description
	Description	PartComment
	Revision Number	PartRevision
	Material	Material

Type	Attribute in Inventor	Key Name in ENF
User Attribute	Title	Title
	Subject	Subject
	Author	Author
	Manager	Manager
	Company	Company
	Category	Category
	Keywords	Keywords
	File Subtype	Document SubType Name
	Stock Number	Stock Number
	Project	Project
	Designer	Designer
	Engineer	Engineer
	Authority	Authority
	Cost Center	Cost Center
	Estimated Cost	Cost
	Status	User Status
	Design State	Design Status
	Checked By	Checked By
	Checked Date	Date Checked
	Eng Approved By	Engr Approved By
	Eng Approved Date	Engr Date Approved
	Mfg Approved By	Mfg Approved By
	Mfg Approved Date	Mfg Date Approved
	WEB Link	Catalog Web Link
	Vendor	Vendor
	Custom	(User definition)



- Supports attributes that belong to following elements only:
 - Assembly (*1)
 - Part
 (*1): Does not support material attributes that belong to this element.

◦ SOLIDWORKS to ENF

Type	Attribute in SOLIDWORK	Key Name in ENF
User Attribute	Custom Properties	(User definition)



- Supports attributes that belong to following elements only:
 - Assembly
 - Part

◦ JT to ENF

Type	Attribute in JT	Key Name in ENF
User Attribute	User Property	(User definition)



- Supports attributes that belong to following elements only:
 - Assembly
 - Part
 - BodyNode

◦ STEP to ENF

Type	Attribute in STEP	Key Name in ENF
System Attribute	id (PRODUCT)	PartNumber
	name (PRODUCT)	PartName
	description (PRODUCT)	Description
	id (PRODUCT_DEFINITION_FORMATIO N)	PartVersion
	description (PRODUCT_DEFINITION_FORMATIO N)	PartComment
	id (PRODUCT_DEFINITION)	PartDefinition
	make_or_buy (PRODUCT_DEFINITION_FORMATIO N_WITH_SPECIFIED_SOURCE)	PartSource
	input file last write time	NativeFileLastWriteTime
	input file size	NativeFileSize
User Attribute	User Defined Attributes	(User definition)



- Supports attributes that belong to following elements only:
 - Assembly
 - Part

◦ STEP AP242 BOM to ENF

Type	Attribute in STEP AP242 BOM	Key Name in ENF
System Attribute	Part/Name/CharacterString (*1)	PartName
	Part/Identifier.id (*1)	PartName or PartNumber
	PartVersion/Identifier.id (*2)	PartRevision
	File path of the source file	NFName
User Attribute (*3)	PropertyValueAssignment	(User definition)

(*1) Specify the mapping rule with a parameter "PartIDMapping".

(*2) Specify the mapping rule with a parameter "PartVersionIDMapping".

(*3) Specify which value to use in case of a key name conflict with a parameter

"XConvertUserProperty".



- Supports attributes that belong to following elements only:
 - Assembly
 - Part

- CADmeister to ENF

Type	Attribute in CADmeister	Key Name in ENF
User Attribute	Parts Attribute	(User definition)



- Supports attributes that belong to following elements only:
 - Object
 - Composite surface and Solid

2. ENF to CAD

- ENF to CATIA V5

Type	Key Name in ENF	Attribute in CATIA V5
System Attribute	PartName	Nomenclature
	PartRevision	Revision
	PartDefinition	Definition
	Description	Description
	PartSource	Source
	PartLayer	Part Layer
	InstanceDescription	Description(Instance)
	Material	Material Name
User Attribute	(User definition)	Other Property



- Supports translation as attributes that belong to following elements only:
 - CATProduct
 - CATPart
 - Shell (Supports material attributes only)
 - Lump (Supports material attributes only)

In NX I-deas to CATIA V5 translation, translates while mapping as follows in Auto Healing process:

- "Part Number" in NX I-deas "PartNumber" in CATIA V5
- "Name" in NX I-deas to "Nomenclature" in CATIA V5

◦ ENF to NX I-deas

Type	Key Name in ENF	Attribute in NX I-deas
System Attribute	PartNumber	Part Number
	PartRevision	Revision
	BinName	Bin Name
	Description	Description
	ChangeHistory	Change History
	Material	Material Name
User Attribute	(User definition)	User Property



- Supports translation as attributes that belong to following elements only:
 - Assembly
 - Part

In CATIA V5 to NX I-deas translation, translates while mapping as follows in Auto Healing process:

- "PartNumber" in CATIA V5 to "Part Number" in NX I-deas
- "Nomenclature" in CATIA V5 to "Name" in NX I-deas

◦ ENF to NX

Type	Key Name in ENF	Attribute in NX
System Attribute	PartNumber	PART_NUMBER (*1)
	PartName	PART_NAME (*1)
	PartRevision	PART_REVISION (*1)
	PartDefinition	PART_DEFINITION (*1)
	Description	PART_DESCRIPTION (*1)
	PartSource	PART_SOURCE (*1)
	ChangeHistory	PART_CHANGE_HISTORY (*1)
	PartComment	PART_COMMENT_NOTE (*1)
	ConfigName	PART_CONFIGURATION (*1)
	PartVersion	PART_VERSION (*1)
	Material	PART_MATERIAL (*1)
	Material	Material
User Attribute	(User definition)	Property

(*1): Creates a user attribute in NX with this name, and translates to it.



- Supports translation as attributes that belong to following elements only:
 - Part (*1)
 - Body (Supports SolidBody and SheetBody only)
 - Face (*1)
 - Edge (*1)
 - Curve (*1)
 - Point (*1)
 - Datum Plane (*1)
- (*1): Does not support material attributes that belong to this element.

- ENF to Creo Parametric

Type	Key Name in ENF	Parameter in Creo Parametric
System Attribute	PartNumber	PARTNUMBER (*1)
	PartName	PARTNAME (*1)
	PartRevision	PARTREVISION (*1)
	PartDefinition	PARTDEFINITION (*1)
	Description	DESCRIPTION (*1)
	PartSource	PARTSOURCE (*1)
	Material	MATERIAL (*1)
	ConfigName	CONFIGNAME (*1)
	NFName	NATIVEFILENAME (*1)
	PartVersion	PARTVERSION (*1)
	TargetFileName	TARGETFILENAME (*1)
	PartLayer	PARTLAYER (*1)
	ChangeHistory	CHANGEHISTORY (*1)
User Attribute	(User definition)	Parameter

(*1): Creates a parameter in Creo Parametric with this name, and translates to it.



- Supports translation as parameters that belong to following elements only:
-Assembly -Part

◦ ENF to Inventor

Type	Key Name in ENF	Attribute in Inventor
System Attribute	Description	Comments
	PartComment	Description
	PartRevision	Revision Number
	Material	Material

Type	Key Name in ENF	Attribute in Inventor
User Attribute	Title	Title
	Subject	Subject
	Author	Author
	Manager	Manager
	Company	Company
	Category	Category
	Keywords	Keywords
	Stock Number	Stock Number
	Project	Project
	Designer	Designer
	Engineer	Engineer
	Authority	Authority
	Cost Center	Cost Center
	Cost	Estimated Cost
	User Status	Status
	Checked By	Checked By
	Date Checked	Checked Date
	Engr Approved By	Eng Approved By
	Engr Date Approved	Eng Approved Date
	Mfg Approved By	Mfg Approved By
	Mfg Date Approved	Mfg Approved Date
	Catalog Web Link	WEB Link
	Vendor	Vendor
	(User definition)	Custom



- Supports translation as attributes that belong to following elements only:
 - Assembly (*1)
 - Part
 (*1): Does not support material attributes that belong to this element.

◦ ENF to SOLIDWORKS

Type	Key Name in ENF	Attribute in SOLIDWORKS
User Attribute	(User definition)	Custom Properties



- Supports translation as attributes that belong to following elements only:
 - Assembly
 - Part
 - Body (*1)
 - Face (*1)
 - Edge Assembly (*1)
 (*1): Translates as "IAttribute" that is accessible via macro or API only

◦ ENF to JT

Type	Key Name in ENF	Attribute in JT
System Attribute	PartName	\$PartName
	PartNumber	\$Nomenclature \$PartNumber (*1)
	NFName	\$NativeFileName
	ConfigName	\$ConfigName
	BinName	\$BinName
	ChangeHistory	\$ChangeHistory
	Description	\$Description
	PartComment	\$PartComment
	PartDefinition	\$PartDefinition
	PartVersion	\$PartVersion
	PartRevision	\$PartRevision
	PartSource	\$PartSource
	PartLayer	\$PartLayer
	NativeFileLastWriteTime	\$NativeFileLastWriteTime
	NativeFileSize	\$NativeFileSize
	InstanceDescription	\$InstanceDescription
	Material	\$Material
User Attribute	(User definition)	User Property

(*1): Translates "PartNumber" in ENF to "\$PartNumber" in JT when translating from CATIA V5, otherwise, translates to "\$Nomenclature" in JT.



- Supports translation as attributes that belong to following elements only:
 - Assembly
 - Part
 - BodyNode (Translates attributes that belong to Geometry Set or user defined features in CATIA V5 as a parameter that belong to BodyNode)

- ENF to STEP

Type	Key Name in ENF	Attribute in STEP
System Attribute	PartNumber	id (PRODUCT)
	PartName	name (PRODUCT)
	Description	description (PRODUCT)
	PartVersion	id (PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE)
	PartComment	description (PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE)
	PartDefinition	id (PRODUCT_DEFINITION)
	PartSource	make_or_buy (PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE)
User Attribute	(User definition)	User Defined Attributes



- Supports translation as attributes that belong to following elements only:
 - Assembly
 - Part

◦ ENF to STEP AP242 BOM

Type	Key Name in ENF	Attribute in STEP AP242 BOM
User Attribute	(User definition)	PropertyValueAssignment



- Supports translation as attributes that belong to following elements only:
 - Assembly
 - Part

◦ ENF to CADmeister (standalone)

Type	Key Name in ENF	Attribute in CADmeister
User Attribute	(User definition)	Parts Attribute



- Supports translation as attributes that belong to following elements only:
 - Object
 - Composite surface and Solid

IV. Attribute Mapping

You can map attributes in ENF to attributes in target CAD formats for your needs. Please refer to 3.3.4, “[Attribute Editor](#)” for the details.

3.4.3. License Management

Licensing of ASFALIS products is managed by a license management tool, Sentinel RMS License Manager. Please refer to "LicenseServer_QuickStartGuide_en.pdf" for how to set up license.

I. Product Code

It requires ASFALIS Component licenses to utilize them. Please refer to "Requirement_ENG.pdf" for the product code ("feature" in Sentinel RMS License Manager) of necessary licenses of each Component.

II. Obtain and Release License

Licenses are checked / obtained / released as follows.



"XX" / "YY" below is to be replaced by the ID of source / target CAD format respectively

1. When running ENF Writer Component (CAD to ENF), "elybatch" runs as Product ID "1XX001".
 1. Run "elybatch" either;
 - With an argument "-p1XX001"
 - As a Scenario <productcode id="1XX001"/>
 2. "elybatch";
 - Checks that a license of "elybatch" (ASF-ELYBATCH) is registered to the license server (It does not consume the license.)
 - Obtains the license for ENF Writer component (ASF-<Code of source CAD>-ENFS)
 3. "elybatch" passes the license (ASF-<Code of source CAD>-ENFS) to ENF Writer Component.
 4. ENF Writer Component runs CAD to ENF translation process consuming that license.
 5. ENF Writer Component notifies "elybatch" on the process completion once completed.
 6. "elybatch" releases the license (ASF-<Code of source CAD>-ENFS).

2. When running a Scenario of [CAD Import] → [Geometry Healing] → [CAD Export], "elybatch" runs as Product ID "1XXYY1".
 1. Run "elybatch" either;
 - With an argument "-p1XXYY1"
 - As a Scenario <productcode id="1XXYY1"/>
 2. "elybatch";
 - Checks that a license of "elybatch" (ASF-ELYBATCH) is registered to the license server (It does not consume the license.)
 - Obtains the license for ENF Writer component (ASF-<Code of source CAD>-ENFS)
 3. ENF Writer Component runs CAD to ENF translation process as a sub process of "elybatch" consuming the license for [CAD Import] (ASF-<Code of source CAD>-ENFS).
 4. ENF Writer Component notifies "elybatch" on the process completion once completed.
 5. "elybatch" releases the license (ASF-<Code of source CAD>-ENFS).
 6. "elybatch" obtains the license for ENF Reader component (ASF-ENFS-<Code of target CAD>).
 7. [Geometry Healing] Component runs healing process as a sub process of "elybatch" consuming the license for ENF Reader (ASF-ENFS-<Code of target CAD>).
 8. [Geometry Healing] Component notifies "elybatch" on the process completion once completed.
 9. "elybatch" releases the license (ASF-ENFS-<Code of target CAD>).
 10. "elybatch" obtains the license for ENF Reader component (ASF-ENFS-<Code of target CAD>).
 11. ENF Reader Component runs ENF to CAD translation process as a sub process of "elybatch" consuming the license for ENF Reader (ASF-ENFS-<Code of target CAD>).
 12. ENF Reader Component notifies "elybatch" on the process completion once completed.
 13. "elybatch" releases the license (ASF-ENFS-<Code of target CAD>).
3. When running ENF Reader Component (ENF to CAD) including annotations and/or attributes, which requires additional license(s) (ASF-ANN/<Code of target CAD>, ASF-ATR/<Code of target CAD> respectively), "elybatch" runs as Product ID "100YY1".

Below is a sample case of ENF to CAD translation including attributes.

Please note that the translation process ends in error when "elybatch" fails to obtain the license of annotation / attribute translation.

1. Run "elybatch" either;
 - With an argument "-p100YY1"
 - As a Scenario <productcode id="100YY1"/>

2. "elybatch";
 - Checks that a license of "elybatch" (ASF-ELYBATCH) is registered to the license server (It does not consume the license.)
 - Obtains the license for ENF Reader component (ASF-ENFS-<Code of target CAD>) as well as the option license for attribute translation (ASF-ATR/<Code of target CAD>)
3. "elybatch" passes the licenses (ASF-ENFS-<Code of target CAD> and ASF-ATR/<Code of target CAD>) to ENF Reader Component.
4. ENF Reader Component runs ENF to CAD translation process consuming those licenses.
5. ENF Reader Component notifies "elybatch" on the process completion once completed.
6. "elybatch" releases the license (ASF-ENFS-<Code of target CAD> and ASF-ATR/< Code of target CAD>).

All rights reserved by Elysium or the original author of this material. The content may not be edited, reproduced, distributed, transmitted, displayed, published, broadcast, sold or lent without the prior permission of the author.