



ASFALISコンポーネント（基本仕様と利用方法）

ASFALISアダプタ, ASFALISオプティマイザ

2020年 8月

株式会社エリジオン

目次

1. 概要	1
1.1. ASFALISアダプタとは	1
1.2. ASFALISオプティマイザとは	1
1.3. 構成	1
1.4. コンポーネントの種類	1
1.5. プロダクトIDについて	3
2. 環境設定：elybatchからの実行方法	5
2.1. 変換を制御するファイル/引数	5
2.2. コンフィグファイル	5
2.2.1. コンフィグファイルキーワード	6
2.2.2. コンフィグファイル内でのキーワードの参照、環境変数の使用について	13
2.2.3. コンフィグファイルバリエーション	15
2.2.4. コンフィグファイル記述例	15
2.3. コマンドライン引数	16
2.3.1. 引数	16
2.3.2. 引数使用例	17
2.4. XMLシナリオ	18
2.4.1. シナリオ書式	18
2.4.2. 実行例	27
2.5. パラメータファイル	30
2.6. エラーコード	30
2.6.1. 注意点	30
3. ASFALIS Component	31
3.1. Adapter	31
3.1.1. CATIA V5 to ENF / ENF to CATIA V5	32
3.1.2. 3DEXPERIENCE to ENF / ENF to 3DEXPERIENCE	34
3.1.3. CATIA V4 to ENF / ENF to CATIA V4	38
3.1.4. NX I-deas to ENF / ENF to NX I-deas	39
3.1.5. Creo Parametric to ENF / ENF to Creo Parametric	41
3.1.6. NX to ENF / ENF to NX	44
3.1.7. Parasolid to ENF / ENF to Parasolid	47
3.1.8. SOLIDWORKS to ENF / ENF to SOLIDWORKS	48
3.1.9. Inventor to ENF / ENF to Inventor	50

3.1.10. Creo Elements/Direct to ENF / ENF to Creo Elements/Direct	51
3.1.11. ACIS to ENF / ENF to ACIS	52
3.1.12. IGES to ENF / ENF to IGES	53
3.1.13. STEP to ENF / ENF to STEP	54
3.1.14. STEP AP242 BOM to ENF / ENF to STEP AP242 BOM	55
3.1.15. JT to ENF / ENF to JT	57
3.1.16. PLM XML to ENF / ENF to PLM XML	58
3.1.17. iCAD to ENF / ENF to iCAD	60
3.1.18. CADmeister to ENF / ENF to CADmeister (standalone)	61
3.1.19. CATIA V5 to ENF / ENF to CATIA V5 (standalone)	63
3.1.20. Creo Parametric to ENF (standalone)	64
3.1.21. NX to ENF (standalone)	64
3.1.22. ENF to STL	65
3.1.23. ENF to XVL	65
3.1.24. ENF to 3D PDF	66
3.1.25. 3DXML to ENF	67
3.2. Healer, etc.	67
3.2.1. TOGO / CADCEUS Healer	67
3.2.2. ENF Polygon	68
3.3. Optimizer	68
3.3.1. PDQChecker	68
3.3.2. Geometry Simplifier	72
3.3.3. CAD Validator	75
3.3.4. Attribute Editor	90
3.3.5. Assembly Editor	98
3.3.6. Interference Checker	106
3.3.7. Polygon Optimizer	108
3.3.8. ENF Editor	110
3.3.9. 3D PDF Editor	127
3.4. Other	128
3.4.1. 3Dアノテーション変換	128
3.4.2. 属性変換	142
3.4.3. ライセンス管理	151
4. 変更履歴	154

1. 概要

1.1. ASFALISアダプタとは

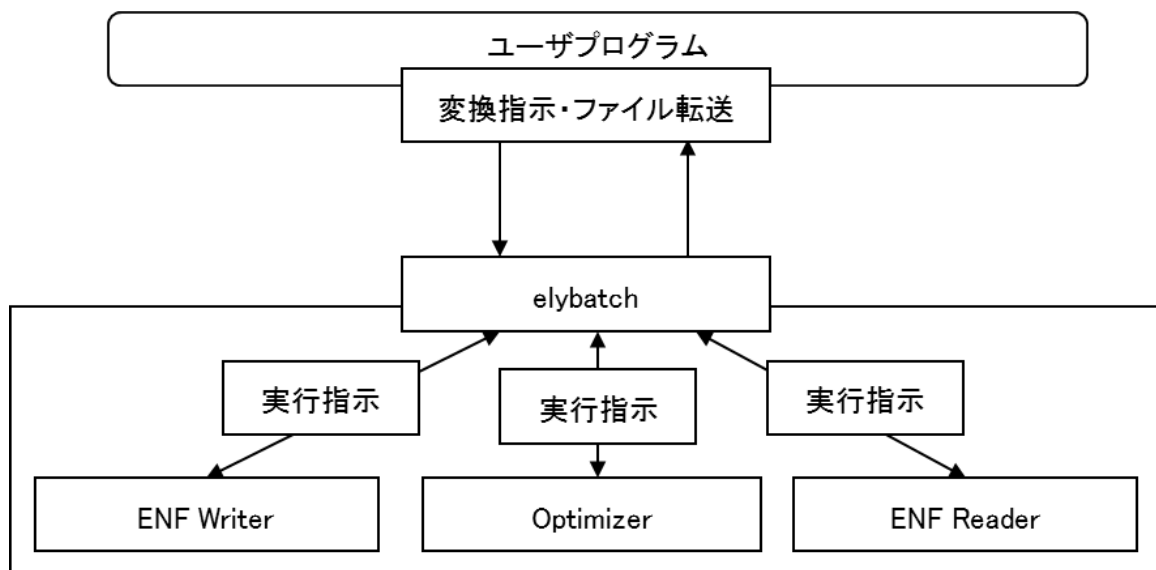
ASFALISアダプタは、各種3次元CADのファイルをENFに変換、逆にENFを各種3次元CADのファイルやカーネル形式のファイルに変換するためのASFALISの中でも最も基本的なコンポーネントです。ASFALISアダプタは、変換の方向によって、3次元CADのファイルをENFに変換する「ENF Writer」(エンフ・ライタ)と、ENFを3次元CADのファイルに変換する「ENF Reader」(エンフ・リーダ)の2つに分類されます。また、それぞれ変換対象の3次元CADによって、全て個別のコンポーネントとして提供されます。

1.2. ASFALISオプティマイザとは

ASFALISオプティマイザは、ENFに対して形状簡略化やPDQチェックといったデータ最適化処理を実行するコンポーネントです。

1.3. 構成

ASFALISアダプタ、ASFALISオプティマイザでは「ENF Writer」「ENF Reader」などのほかにそれらを稼動するためのコントロールプログラム「elybatch」が提供されます。



ENF Writer/ENF Readerを直接実行するのではなく、このelybatchを実行することでENF Writer/ENF Readerを実行することになります。

1.4. コンポーネントの種類

利用可能なASFALISアダプタの種類は以下のとおりです。

- ASFALIS アダプタ

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(UFUNC) to ENF	ENF to NX(UFUNC)
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
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) CATIA V5からENFに直接変換するのではなく、途中でCADdoctor形式を経由してENFに変換されます。

(*2) ENFからCATIA V5に直接変換するのではなく、途中でCADdoctor形式を経由してCATIA V5に変換されます。

ASFALISアダプタと同時に利用可能なASFALISオプティマイザの種類は以下の通りです。

- ASFALIS オプティマイザ

PDQ checker	PDQチェッカ
Geometry Simplifier	形状簡略化

CAD Validator	CAD比較
Attribute Editor	属性編集
Assembly Editor	アセンブリ編集
Interference Checker	干渉チェッカ
Polygon Optimizer	ポリゴン最適化
ENF Editor	ENF編集

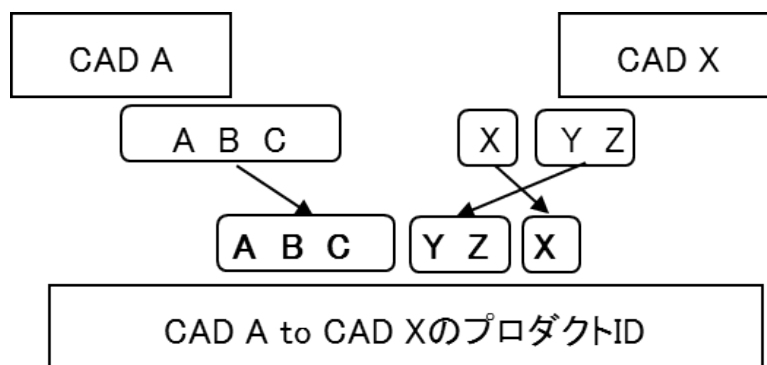
1.5. プロダクトIDについて

各CADはそれぞれ固有のIDが割り振られており、そのIDを組み合わせることでプロダクトIDが作られます。各CADのIDは以下の通りです。

CAD	CAD ID
ENF	100
ACIS	102
NX I-deas	103
Creo Parametric	104
Parasolid	105
CATIA	106
SOLIDWORKS	109
Creo Elements/Direct Modeling	112
NX(UFUNC)	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

CAD	CAD ID
STEP AP242 BOM	175
3D PDF	176
STL	181
JT	184
PLM XML	194
3DEXPERIENCE	195
STEP	198
IGES	199

CAD A(CAD IDがABC) to CAD X(CAD IDがXYZ)の場合はABCYZXがプロダクトIDとなります。



(例)

- Creo Parametric to NX I-deasの場合、Creo Parametricが104、NX I-deasが103なので104031 となります。
- CATIA V5 to Parasolidの場合、CATIA V5が117、Parasolidが105なので117051 となります。
- NX(UFUNC) to ENFの場合、NX(UFUNC)が114、ENFが100なので114001 となります。

2. 環境設定 : elybatchからの実行方法

2.1. 変換を制御するファイル/引数

コンフィグファイル	変換で使用するコンポーネントやCAD本体などへのパス、変換前のライセンスチェック設定等、変換環境の基本的な情報を指定するファイルです。
引数	入出力ファイル、パラメータファイル、ログファイル、変換の作業フォルダ等、変換毎の情報は引数で指定します。 引数でXMLシナリオを指定し、入出力ファイルのパス等はXMLシナリオ内で指定することも可能です。
XMLシナリオ	変換処理順、およびその際の入出力ファイルの情報等を指定するファイルです。
パラメータファイル	変換時にコンポーネントの細かい設定を指定するファイルです。引数またはXMLシナリオにファイルパスを記述します。

引数を使用した変換制御では、基本的にはCAD to CADの変換を行います。引数の指定によりヒーリング後で処理を止める、ヒーリングのみをおこなうなどの設定も可能ですが、PDQCheck以外のOptimizerを実行することはできません。XMLシナリオを使用した変換制御では、より柔軟にコンポーネントの呼び出し順を制御することができます。

2.2. コンフィグファイル

コンフィグファイルには、変換で使用するコンポーネントやCAD本体などへのパス、変換前のライセンスチェック設定等、変換環境の基本的な情報を指定します。コントロールプログラム elybatch はこのファイルの情報を元にコンポーネントを呼び出します。指定する際には、予め規定されているパラメータの名称 (以下「キーワード」) とそれらに指定する実際の値 (パスやポート番号等) の組を記載します。以下の例の通り、コンフィグファイルの最初から1番目の「PRODUCT <数字>」の前の行までを共通セクション、「PRODUCT <数字>」の下をプロダクトセクションと呼びます。

```
#共通セクション
ESERVER          C:\Asfalisp
PORT_NUM         5093
LIC_SERVER       XXX
TESSELLATOR      $ESERVER\tools\win

#プロダクトセクション
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
```

- 各キーワードについては「[2.2.1. “コンフィグファイルキーワード”](#)」を参照してください。

- 共通セクションの ESERVER、PORT_NUM、LIC_SERVER、プロダクトセクションの PRODUCT、BATCH_STATUS は必須項目です。
- #で始まる行はコメント行として無視されます。
- キーワードの前もしくは値の後ろに空白文字があると動作しません。
- キーワードと値の間には1つ以上の空白文字 (半角スペースまたはタブ) が必要です。
- 空白文字だけの行は無視されます。
- 全角スペースは空白文字として使用できません。
- 文字コードは ANSI コードページを使用してください。
- 改行コードは LF もしくは CR+LF を使用してください。

2.2.1. コンフィグファイルキーワード

- 共通セクション

ESERVER	
説明	ASFALIS モジュールが置かれているルートフォルダを指定する。
記述例	ESERVER C:\Asfalis
PORT_NUM	
説明	ライセンスサーバのポート番号を記入
記述例	PORT_NUM 5093
LIC_SERVER	
説明	ライセンスサーバのマシン名 (IPアドレスも可)
記述例	LIC_SERVER XXXX
TESSELLATOR	
説明	共有モジュールへのパス
記述例	TESSELLATOR \$ESERVER\tools\win
INTRCHECK	
説明	干渉チェッカモジュールへのパス
記述例	INTRCHECK \$ESERVER\tools\win\elyIntrCheck\IntrCheck
EXTEND_ERROR_INFO	
説明	拡張エラーのリターンコードを出力 (0/1/2) 1以上を設定すると1000以上のリターンコードが返る 1を設定すると1000-1010まで、2を設定すると全種類のリターンコードが出力される
記述例	EXTEND_ERROR_INFO 2
SCENARIO_FILE	

説明	ElybatchシナリオXMLファイルへのパス
記述例	SCENARIO_FILE C:\Asfalis\scenario\scenario.xml
SCENARIO_LOG	
説明	Elybatchシナリオログへのパス
記述例	SCENARIO_LOG C:\Asfalis\log\scenariolog.xml
SCENARIO_PROGRESS	
説明	プログレス情報を出力 (0/1) コンポーネントの進捗状況をprogressというファイルに出力する シナリオコントロールでの実行時のみの機能
記述例	SCENARIO_PROGRESS 1
NUM_OF_PROCESSES	
説明	<p>変換に使用する最大プロセス数を指定する。 値を指定しない場合はアダプタ毎に適切な値が使用される。 0を指定した場合は使用可能な最大プロセス数が使用される。 以下のコンポーネントに有効</p> <ul style="list-style-type: none"> • NX, CATIA V5, JT のENF Reader <ul style="list-style-type: none"> -LEGACY_MODE を設定していないか0 を設定した場合に有効 -プロダクトセクションにも指定した場合はそちらが優先される • PLM XML, STEP AP242 BOM の ENF Writer <ul style="list-style-type: none"> -LEGACY_MODE を設定していないか0 を設定した場合に有効 -プロダクトセクションにも指定した場合はそちらが優先される • CAD Validator
記述例	NUM_OF_PROCESSES 4
BATCH_TIME_OUT	
説明	変換処理のタイムアウト時間を分単位で指定する。指定しない場合は360分(6時間)が設定される。
記述例	BATCH_TIME_OUT 12

• プロダクトセクション(プロダクト共通)

PRODUCT	
説明	プロダクトID
記述例	PRODUCT 117001
FROM_CAD_START_CMD	
説明	ENF Writerが利用するCADの起動コマンド NX I-DEAS、Creo ParametricなどCADを利用するENF Writerを利用する際は必須

記述例	FROM_CAD_START_CMD C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
PATH_FROM_TRANSLATOR	
説明	環境変数PATHに追加するパスを記述。ENF Writerの実行時に利用される。CATIA V5などCADを利用するENF Writerを利用する際に設定が必要なものがある。
記述例	PATH_FROM_TRANSLATOR C:\Program Files\Dassault Systemes\B28\intel_a\code\bin (* 改行は不要)
FROM_TRANSLATOR	
説明	ENF Writerのモジュールへのパス
記述例	FROM_TRANSLATOR \$ESERVER\CaaExe\win\B28
ADEK_DIR	
説明	ヒーリングモジュールへのパス
記述例	ADEK_DIR \$ESERVER\tools\win
TO_TRANSLATOR	
説明	ENF Readerのモジュールへのパス
記述例	TO_TRANSLATOR \$ESERVER\CaaExe\win\B28
TO_CAD_START_CMD	
説明	ENF Readerが利用するCADの起動コマンド NX I-DEAS、Creo ParametricなどCADを利用するENF Readerを利用する際は必須
記述例	TO_CAD_START_CMD C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
PATH_TO_TRANSLATOR	
説明	環境変数PATHに追加するパスを記述。ENF Readerの実行時に利用される。CATIA V5などCADを利用するENF Readerを利用する際に設定が必要なものがある
記述例	PATH_TO_TRANSLATOR C:\Program Files\Dassault Systemes\B28\intel_a\code\bin (* 改行は不要)
BATCH_STATUS	
説明	elybatch用フラグ。常に記載してください。
記述例	BATCH_STATUS
NUT_CHECK_DIR	
説明	ASFALISオプティマイザのPDQチェッカのモジュールのディレクトリ
記述例	NUT_CHECK_DIR \$ESERVER\tools\win
LM_FROM_TRANSLATOR, LM_TO_TRANSLATOR	

説明	Flexlmでライセンス管理されているCADのライセンスをチェックするために利用される。lmutilという名前の実行ファイルのディレクトリを指定する。 ライセンスチェックを利用するにあたっては、 LM_FROM_LICENSE, LM_TO_LICENSE, LM_FROM_FEATURE, LM_TO_FEATURE と同時に設定することが必要。
記述例	LM_FROM_TRANSLATOR C:\Program Files\PTC\Creo 4.0\F000\Common Files\x86_win\obj (* 改行は不要)
LM_FROM_LICENSE, LM_TO_LICENSE	
説明	Flexlmのライセンスサーバが起動しているホストとポートを指定する。 複数のサーバが起動している場合も指定が可能。
記述例	LM_FROM_LICENSE 7788@hostname1;7788@hostname2
LM_FROM_FEATURE, LM_TO_FEATURE	
説明	CADのライセンスが設定されているFEATUREを指定する。 FEATUREの複数指定が可能。10個まで指定が可能。
記述例	LM_FROM_FEATURE solid_modeling;gateway
LM_TO_FEATURE_OPERATOR, LM_FROM_FEATURE_OPERATOR	
説明	Flexlmのライセンスチェックの判定方法を指定する。"AND"または"OR"を指定。 <ul style="list-style-type: none"> • AND : LM_FROM_FEATURE, LM_TO_FEATUREに指定した全てのFEATUREのライセンスに空きがある場合にライセンス取得可能であると判定する。 • OR : LM_FROM_FEATURE, LM_TO_FEATUREに指定したFEATUREのうち1つ以上のライセンスに空きがある場合にライセンス取得可能であると判定する。 何も指定しない場合はORの動作となる。 • LM_FROM_FEATURE, LM_TO_FEATURE, LM_FROM_LICENSE, LM_TO_LICENSE, LM_FROM_TRANSLATOR, LM_TO_TRANSLATORと同時に設定することが必要。
記述例	LM_FROM_FEATURE_OPERATOR AND
LM_LICENSE_WAIT_COUNT	
説明	LM_FROM_LICENSE, LM_FROM_FEATURE, LM_TO_LICENSE, LM_TO_FEATURE で指定されたFlexlmのライセンスチェックを再実行する回数。 指定しなければ0回が設定される。
記述例	LM_LICENSE_WAIT_COUNT 10
LM_LICENSE_WAIT_INTERVAL	
説明	LM_LICENSE_COUNTが1以上の場合有効。 LM_FROM_LICENSE, LM_FROM_FEATURE, LM_TO_LICENSE, LM_TO_FEATURE で指定されたFlexlmのライセンスチェックを再実行する間隔。 指定しなければ30秒が設定される。
記述例	LM_LICENSE_WAIT_INTERVAL 30
LICENSE_WAIT_COUNT	

説明	<CAD> to ENF, オプティマイザ, ENF to <CAD>のエリジオンライセンスのチェックを再実行する回数。指定しなければ0回が設定される。
記述例	LICENSE_WAIT_COUNT 5
LICENSE_WAIT_INTERVAL	
説明	LICENSE_WAIT_COUNTが1以上の場合有効。 <CAD> to ENF, オプティマイザ, ENF to <CAD>のエリジオンライセンスのチェックを再実行する間隔。指定しなければ30秒が設定される。
記述例	LICENSE_WAIT_INTERVAL 30
LICENSE_DETAIL	
説明	エリジオンライセンスのエラーが起きた際にelybatchが詳細なエラーコードを返す。
記述例	LICENSE_DETAIL
BATCH_TIME_OUT	
説明	変換処理のタイムアウト時間を分単位で指定する。指定しない場合は360分(6時間)が設定される。
記述例	BATCH_TIME_OUT 120
LEGACY_MODE	
説明	旧アダプタを使用する / しないを指定する。(0/1) 0: アダプタを使用する 1: 旧アダプタを使用する デフォルトは、CADおよびENF Writer / Readerごとに異なる
記述例	LEGACY_MODE 1
NUM_OF_PROCESSES	
説明	変換に使用する最大プロセス数を指定する。 値を指定しない場合はアダプタ毎に適切な値が使用される。 0を指定した場合は使用可能な最大プロセス数が使用される。 以下の条件を満たす場合に有効 - LEGACY_MODEを設定していないか0を設定した場合 - NX, CATIA V5, JTのENF Reader - PLM XML, STEP AP242 BOM のENF Writer
記述例	NUM_OF_PROCESSES 4

- プロダクトセクション(プロダクト依存)

3DEX_Repository	
説明	3DEXPERIENCE用ENF Writer/Readerで利用する。 変換時に使用するレポジトリ名を指定する。 指定しない場合は PLM1 が使われる。
記述例	3DEX_Repository PLM1
3DEX_Server	
説明	3DEXPERIENCE用ENF Writer/Readerで利用する。 3DSpaceのURLを指定する。 フォーマットは <protocol>://<HostName>:<Port>/<RootURI>
記述例	3DEX_Server https://servermachine.mycompany.co.jp:8080/3dspace
3DEX_LoginTicket	
説明	3DEXPERIENCE用ENF Writer/Readerで利用する。 ログインチケット文字列を指定する。
記述例	3DEX_LoginTicket NEE3QkYxNzY4N0MyNDJDMEFEMzVFNUU5NDE4MEEyNUV8ZGV2MjAxN3h8ZGV2 (* 改行は不要)
3DEX_PLMType	
説明	3DEXPERIENCE用ENF Writer/Readerで利用する。 PLM Typeを指定する。 指定しない場合は VPMReference が使用される。
記述例	3DEX_PLMType VPMReference
3DEX_EnvPath	
説明	3DEXPERIENCE用ENF Writer/Readerで利用する。 変換時に使用する 3DEXPERIENCE の環境ファイルをフルパスで指定する。 指定しない場合はデフォルトの環境ファイルが使用される。
記述例	3DEX_EnvPath C:\Program Files\Dassault Systemes\B420\CATEnv\Env.txt
COLOR_DEF_FILE	
説明	NX(UFUNC)用ENF Writerで利用する。 NXの色定義ファイル (*.cdf) をフルパスで指定する。
記述例	COLOR_DEF_FILE D:\data\ugcolor_sample.cdf
WIDTH_DEF_FILE	
説明	NX(UFUNC)用ENF Writerで利用する。 線幅マッピングファイル (*.txt) をフルパスで指定する。
記述例	WIDTH_DEF_FILE D:\data\width_def_file.txt
PARASOLID_SCHEMA	
説明	Parasolid用ENF Writer/Readerで利用する。 Schemaファイルのディレクトリを指定する
記述例	PARASOLID_SCHEMA \$ESERVER\ParasolExe\schema

OPT_FROM_TRANSLATOR	
説明	Creo Parametric用ENF Writerで利用する。
記述例	OPT_FROM_TRANSLATOR pro_wait
OPT_TO_TRANSLATOR	
説明	Creo Parametric用ENF Readerで利用する。
記述例	OPT_TO_TRANSLATOR pro_wait
PRO_NMSD_FINISH	
説明	Creo Parametric用 ENF Writer / Reader で利用する。 Elybatchが終了する時に、nmsd.exe (Creo Parametricの起動監視デーモン) も終了する。
記述例	PRO_NMSD_FINISH
PRO_PROD_DCU_FINISH	
説明	Creo Parametric用 ENF Writer / Reader で利用する。 Elybatchが終了する時に、prod_dcu.exe (PTC Quality Agentに必要なシステム情報のXMLファイルを作成するプロセス) も終了する。(0/1)
記述例	PRO_PROD_DCU_FINISH 1
CATIAV5_EnvPath	
説明	CATIA V5 用ENF Writer/ Reader で利用する。 変換時に使用するCATIA V5の環境ファイルをフルパスで指定する。 指定しない場合はデフォルトの環境ファイルが使用される。
記述例	CATIAV5_EnvPath C:\ProgramData\DassaultSystemes\CATEnv\CATIA.V5-6R2018.B28_MyEnv.txt (* 改行は不要)
CAA_WAIT_COUNT	
説明	CATIA V5用ENF Writer/ Readerで利用する。 CATIA V5の起動に時間がかかることがあるため、起動待ち処理を行う。 その回数を設定する。指定しなければ30回が設定される。
記述例	CAA_WAIT_COUNT 30
CAA_WAIT_INTERVAL	
説明	CATIA V5用ENF Writer/ Reader で利用する。 CAA_WAIT_COUNTで指定した試行の間隔。指定しなければ5秒が設定される。
記述例	CAA_WAIT_INTERVAL 10
CAA_LICENSE_WAIT_COUNT	
説明	CATIA V5 ENF Writer/ Reader でライセンスが取得できず実行エラーになった場合、変換の再試行を行う。 CAA_LICENSE_WAIT_COUNTで再試行の回数を指定する。 指定しない場合は0回が設定され、再試行は行われない。
記述例	CAA_LICENSE_WAIT_COUNT 10
CAA_LICENSE_WAIT_INTERVAL	

説明	CAA_LICENSE_WAIT_COUNTで指定された再試行の間隔を指定する。 単位は秒。指定しない場合は30秒が設定される。
記述例	CAA_LICENSE_WAIT_INTERVAL 30
TEMPLATE_ASSEMBLY	
説明	CATIA V5 ENF Reader, Creo Parametric ENF Reader, NX ENF Reader で利用 する。 テンプレートとして用いるアセンブリファイルをフルパスで指定する。
記述例	TEMPLATE_ASSEMBLY C:\data\template\template.asm.1
TEMPLATE_PART	
説明	CATIA V5 ENF Reader, Creo Parametric ENF Reader, NX ENF Readerで 利用する。 レートとして用いるパートファイルをフルパスで指定する。
記述例	TEMPLATE_PART C:\data\template\template.prt.1

2.2.2. コンフィグファイル内でのキーワードの参照、環境変数の使用について

コンフィグファイル内で、コンフィグキーワードや環境変数を外部変数として参照することが可能です。参照できる変数は以下のとおりです。

- コンフィグファイルの共通セクションに書かれたESERVER
- elybatch実行前に設定されている任意の環境変数

「\$変数」と記述すると、変数の値が参照されます。

使用例:

- 環境変数設定

```
CADVER=21
```

- コンフィグファイル内 共通セクションの記述

```
ESERVER          C:\Asfalıs\bin
```

- コンフィグファイル内 プロダクトセクションの記述

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

上記のように指定した場合、下記の指定と同じ意味になります。

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

2.2.3. コンフィグファイルバリエーション

elybatchの引数 -T 指定を行うことで、同じプロダクトIDで別設定を読み出すことが可能です。例えば複数の CAD バージョンの設定を1つのファイルに作成することができます。

- コンフィグファイル例

```
PRODUCT          117001
FROM_TRANSLATOR   C:\Asfalisp\bin\CaaExe\win\B21
PATH_FROM_TRANSLATOR C:\Program Files\Dassault Systemes\B21\intel_a\code\bin
(中略)

PRODUCT          117001 22
FROM_TRANSLATOR   C:\Asfalisp\bin\CaaExe\win\B22
PATH_FROM_TRANSLATOR C:\Program Files\Dassault Systemes\B22\intel_a\code\bin
(中略)

PRODUCT          117001 20
FROM_TRANSLATOR   C:\Asfalisp\bin\CaaExe\win\B20
PATH_FROM_TRANSLATOR C:\Program Files\Dassault Systemes\B20\intel_a\code\bin
(中略)
```

- 引数-Tを指定しないで変換した場合、CATIA V5R21での変換が行われます。
- 以下のように指定して変換した場合、CATIA V5R20での変換が行われます。

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

2.2.4. コンフィグファイル記述例

- ASFALISアダプタを利用してCATIAV4 to NX I-deas変換を行う場合

```
ESERVER          C:\Asfalisp
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
```

- ASFALISアダプタを利用してCreo Parametric to IGES変換を行う場合

```

ESERVER          C:\Asfalisp
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. コマンドライン引数

2.3.1. 引数

elybatchのコマンドライン引数は以下のとおりです。

```

elybatch.exe -f<コンフィグファイル名> -p<プロダクトID> -w<ワークディレクトリ> -
i<入力ファイル名> -o<出力ファイル名> -u<ログファイル名> -c<パラメータファイル名>
-X<XMLログファイル名>

```

-f	【引数設定必須】 コンフィグファイル名(絶対パス またはワークディレクトリからの相対パス)
-w	【引数設定必須】 ワークディレクトリ(絶対パス)。UNCパスはサポート対象外。
-p	【シナリオを使用しない場合には引数設定必須。 シナリオ実行時には、シナリオ内で指定することで引数の省略が可能。】 プロダクトID オプティマイザ実行時には引数"-p"を与えないこと。
-i	【シナリオを使用しない場合には引数設定必須。 シナリオ実行時には、シナリオ内で指定することで引数の省略が可能。】 入力ファイル名(*1) (絶対パス またはワークディレクトリからの相対パス) CADファイル名またはENFファイル名(*.enf)を指定する。
-o	【シナリオを使用しない場合には引数設定必須。 シナリオ実行時には、シナリオ内で指定することで引数の省略が可能。】 出力ファイル名(*1) (絶対パス またはワークディレクトリからの相対パス) CADファイル名を指定する。 停止オプションをつかう場合はENFファイル名(*.enf)を指定する。
-u	テキスト形式のログファイル名 (絶対パス またはワークディレクトリからの相対パス) 引数-uとXMLシナリオの<logfile>タグに異なるファイル名が指定されている場合、 シナリオの各コンポーネントの処理中は<logfile>で指定されたファイルにログが出力される。
-X	XML形式のログファイル名 (絶対パスまたは、ワークディレクトリからの相対パス)

-c	パラメータファイル名 (絶対パスまたは、ワークディレクトリからの相対パス)
-s	停止オプション。-b, -aと同時に使用する。
-b	停止オプション。 "-s -b"を指定すると変換処理がヒーリング処理前で終わりENFファイルが出力される。 その際"-o"はENFの名前"XXX.enf"を指定すること。
-a	停止オプション。 "-s -a"を指定すると変換処理がヒーリング処理後に終わりENFファイルが出力される。 その際"-o"はENFの名前"XXX.enf"を指定すること。
-k	ヒーリングスキップオプション。 "-i"でENFの名前"XXX.enf"を指定すること。ヒーリング処理をせずに変換処理を行う。
-n	PDQチェッカを実行する。(*2)
-t	変換処理のタイムアウト時間を分単位で指定する。指定しない場合は360分(6時間)。 コンフィグファイルにBATCH_TIME_OUTの指定がある場合は"-t"の指定が優先される。
-T	コンフィグファイルバリエーションの指定。1以上の整数を指定する。コンフィグファイルバリエーションについては「 2.2.3, “コンフィグファイルバリエーション” 」を参照のこと
-S	シナリオXMLファイル名 (絶対パス またはワークディレクトリからの相対パス)
-U	シナリオログファイル名 (絶対パス またはワークディレクトリからの相対パス)

(*1) ENF Writer/Readerによって挙動が変わります。詳しくは「[3.1, “Adapter”](#)」の項を参照してください。

(*2) 停止オプション、ヒーリングスキップオプションによって挙動が変わります。詳しくは、「[3.3.1, “PDQChecker”](#)」の項を参照してください。

2.3.2. 引数使用例

1.	CATIAV4ファイル(test.model)からParasolidファイル(test.x_t)を作成する場合 elybatch.exe -felybatch.cfg -p106051 -itest.model -otest.x_t -wC:\elybatch\work
2.	Creo Parametric ファイル(test.prt)から、Parasolidファイル(test.x_t)を作成し、テキストログファイル(test.log)を作成する場合 elybatch.exe -felybatch.cfg -p104051 -itest.prt -otest.x_t -utest.log -wC:\elybatch\work
3.	CATIAV4ファイル(test.model)から、NX I-DEASへの変換処理でヒーリング処理をしないENFファイル(test.enf)を出力する場合 elybatch.exe -felybatch.cfg -p106031 -itest.model -otest.enf -s -b -wC:\elybatch\work
4.	CATIAV4ファイル(test.model)から、NX I-DEASへの変換処理でヒーリング処理をしたENFファイル(test_cl.enf)を出力する場合 elybatch.exe -felybatch.cfg -p106031 -itest.model -otest_cl.enf -s -a -wC:\elybatch\work
5.	NX I-DEASから出力された中間ファイル(test.enf)でヒーリング処理をされていないものからCATIAV4ファイル(test.model)を作成する場合 elybatch.exe -felybatch.cfg -p103061 -itest.enf -otest.model -wC:\elybatch\work

6.	NX I-DEASから出力されたENFファイル(test.enf)でヒーリング処理をされていないものからCATIAV4ファイルを作成するためにヒーリングされたENFファイル(test_cl.enf)を作成する場合
	elybatch.exe -felybatch.cfg -p103061 -itest.enf -otest_cl.enf -s -a -wC:\elybatch\work

2.4. XMLシナリオ

コンポーネントの呼び出し順を外部ファイルに定義することで、elybatch がその順番(シナリオ)に沿って、各コンポーネントを呼び出します。この外部ファイルはXML形式で記述します。シナリオから呼び出す事ができるコンポーネントは以下の通りです。

CAD2ENF	CADファイルから中間ファイルに変換する (ENF Writer)
ENF2CAD	中間ファイルからCADファイルに変換する (ENF Reader)
DEK	ヒーリングを実行する
PDQ	PDQチェックを実行する
SIMP	形状簡略化を実行する
GEOMDIFF	CADモデルどうしの比較を実行する
ATTREDIT	属性編集を実行する
ASMEDIT	アセンブリ編集を実行する
INTRCHK	干渉チェックを実行する
POLYGON	ポリゴン要素を作成する
PLYGNOPTIMIZE	ポリゴン最適化を実行する
ENFEDITOR	ENF編集を実行する
DSCHECK	DFM Studio Checkerを実行する
DSREPORT	DFM Studio Reporterを実行する

2.4.1. シナリオ書式

I. シナリオ書式

文字コードの指定はありませんが、XML の書式に従いファイルの文字コードを宣言するようにしてください。

相対パスについては、作業フォルダを基点に指定してください。

- XMLの構造
(*各コンポーネントの詳細な指定方法は、下記を参照してください。)

Header
ScenarioList
Scenario

各コンポーネントの呼び出しタグ

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

- 外部参照

シナリオXMLでは、引数や環境変数を外部変数として指定する事が可能です。

```
${ 変数のキー } = 変数の値
```

\${ } で指定するキーは全て大文字で定義する必要があります。

- 予約語

以下の変数は予約語として定義します。

環境変数に指定された変数のキーと予約語が重複した場合は、変数のキーが優先されます。

\${INPUTFILE}	-i で指定されているファイル (パス含む)
\${INPUTPATH}	-i で指定されているファイル名までのパス
\${INPUTNAME}	-i で指定されているファイル名
\${INPUTTEXT}	-i で指定されているファイルの拡張子 (「.」ピリオド除く)

```
${INPUTFILE} = ${INPUTPATH}\${INPUTNAME}.${INPUTTEXT}
```

\${OUTPUTFILE}	-o で指定されているファイル (パス含む)
\${OUTPUTPATH}	-o で指定されているファイル名までのパス
\${OUTPUTNAME}	-o で指定されているファイル名
\${OUTPUTTEXT}	-o で指定されているファイルの拡張子 (「.」ピリオド除く)

\${LOGFILE}	-u で指定されているファイル (パス含む)
\${PARAMETERFILE}	-c で指定されているファイル (パス含む)
\${WORKDIR}	-w で指定されているワークディレクトリ
\${CONFIGFILE}	-f で指定されているファイル (パス含む)
\${XMLLOGFILE}	-X で指定されているファイル (パス含む)

\${PRODUCTID}	-p で指定されているproductID
\${PREPRODUCTID}	-p で指定されている 元CAD側のproductID ("117"XXX)
\${POSTPRODUCTID}	-p で指定されている 相手CAD側のproductID (XXX "171")

\${TIMEOUT}	-t で指定されている タイムアウト時間
-------------	----------------------

- 引数との整合性

シナリオXMLファイルを指定した場合、引数の指定は外部参照として利用され、従来の引数の利

用はできません。また、シナリオXMLを指定した場合は、-s, -b, -a, -k, -n は無視されます。

II. シナリオタグ

以下のシナリオタグをシナリオXMLに定義することが可能です。

セクション: CAD2ENF

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス	M
<outputfile>	出力ファイル	M	path	絶対パス/相対パス指定	M
<productcode>	プロダクト	M	id	IDを指定 ENF Writerの場合は 必ず xxx001	M
<workdir>	ワークフォルダ	M	path	絶対パス指定 UNCパスは非サポート	M
<parameterfile>	パラメータファイル	O	path	絶対パス/相対パス指定	M
<logfile>	テキスト形式のログ ファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファ イル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: ENF2CAD

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	M	path	絶対パス/相対パス指定	M
<productcode>	プロダクト	M	id	IDを指定 ENF Readerの場合は 必ず 100xxx	M
<workdir>	ワークフォルダ	M	path	絶対パス指定 UNCパスは非サポート	M
<parameterfile>	パラメータファイル	O	path	絶対パス/相対パス指定	M
<logfile>	テキスト形式のログ ファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファ イル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: DEK

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	M	path	絶対パス/相対パス指定	M
<productcode>	プロダクト	M	id	IDを指定	M
<workdir>	ワークフォルダ	M	path	絶対パス指定 UNCパスは非サポート	M
<parameterfile>	パラメータファイル	O	path	絶対パス/相対パス指定	M
			productcode	<productcode>に指定されているパラメータを実行する	O
<logfile>	テキスト形式のログファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファイル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: PDQ

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<csvfile>	出力ファイル(csv形式)	M	path	絶対パス/相対パス指定	M
<outputfile>	チェック結果を含んだ ENFを出力	O	path	絶対パス/相対パス指定	M
<formatfile>	フォーマットファイル	M	path	絶対パス/相対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス UNCパスは非サポート	M
<parameterfile>	チェック項目用の設定ファイル	M	path	絶対パス/相対パス指定	M
<logfile>	ログファイル	O	path	セットしない場合は stdoutに出力	M
<xmllogfile>	XML形式のログファイル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: SIMP

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	M	path	絶対パス/相対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス UNCパスは非サポート	M
<parameterfile>	パラメータファイル	M	path	絶対パス/相対パス指定	M
<logfile>	テキスト形式のログ ファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファ イル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: GEOMDIFF

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<sourcefile>	比較元ファイル	M	path	絶対パス/相対パス指定	M
<targetfile>	比較するファイル	M	path	絶対パス/相対パス指定	M
<diffresultfile>	出力比較結果ファイ ル	O *1	path	絶対パス/相対パス指定	M
<parameterfile>	パラメータ	O	path	絶対パス/相対パス指定	M
<logfile>	テキスト形式のログ ファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファ イル	O *1	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M
<workdir>	ワークフォルダ	M	path	絶対パス UNCパスは非サポート	M
<xmlreportfile>	形状比較レポートフ ァイル (XML形式) *2, *5	O	path	絶対パス/相対パス指定	M
<html3dreportfil e>	比較レポートファイ ル (3D HTML形式) *3, *5	O	path	絶対パス/相対パス指定	M
<pdf3dreportfile >	比較レポートファイ ル (3D PDF形式) *4	O	path	絶対パス/相対パス指定	M

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<diffanalyzer>	ビューア形式レポート (exe 形式) *6	O	path	絶対パス/相対パス指定	M

- *1 どちらかの指定が必要。
- *2 パラメータファイルでCreateReport=1を指定しない場合、<xmlreportfile>は無視されます。
パラメータファイルでCreateReport=1を指定して<xmlreportfile>を指定しない場合、
<diffresultfile>もしくは<xmllogfile>の名前を元にして名前を自動決定します。
<xmlreportfile>で指定したファイルが既に存在する、または拡張子を除いたファイル/フォルダが
既に存在する場合、レポートファイルは保存されません。
- *3 パラメータファイルでCreate3DReport=1を指定しない場合、<html3dreportfile>は無視され
ます。
パラメータファイルでCreate3DReport=1を指定して<html3dreportfile>を指定しない場合、
<diffresultfile>もしくは<xmllogfile>の名前を元にして名前を自動決定します。
<html3dreportfile>で指定したファイルが既に存在する、または拡張子を除いたファイル/フォル
ダが既に存在する場合、レポートファイルは保存されません。
- *4 パラメータファイルでCreate3DPdfReport=1を指定しない場合、<pdf3dreportfile>は無視さ
れます。
パラメータファイルでCreate3DPdfReport=1を指定して<pdf3dreportfile>を指定しない場合、
<diffresultfile>もしくは<xmllogfile>の名前を元にして名前を自動決定します。
<pdf3dreportfile>で指定したファイルが既に存在する場合、レポートファイルは保存されませ
ん。
- *5 <xmlreportfile> と <html3dreportfile> に拡張子以外が同じ名前のファイルパスを指定するこ
とはできません。形状比較レポートファイル(XML形式) と形状比較レポートファイル(3D HTML
形式)はいずれもサブフォルダを作りますが、そのフォルダ名が重なるためです。
- *6 パラメータファイルでCreateDiffAnalyzer=1を指定しない場合<diffanalyzer>は無視されま
す。 パラメータファイルでCreateDiffAnalyzer=1を指定して<diffanalyzer>を指定しない場
合、<diffresultfile>もしくは<xmllogfile>の名前を元にして名前を自動決定します。
<diffanalyzer>で指定したファイルが既に存在する場合、レポートファイルは保存されません。

セクション: ATTREDIT

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	O	path	絶対パス/相対パス指定	M
<bomfile>	属性のリストファイ ル	O	path	絶対パス/相対パス指定	M
<mapfile>	マッピングファイル	O	path	絶対パス/相対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス UNCパスは非サポート	M

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<logfile>	テキスト形式のログファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファイル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: ASMEDIT

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	M	path	絶対パス/相対パス指定	M
<parameterfile>	パラメータファイル	M	path	絶対パス/相対パス指定	M
<namelist>	削除対象リスト	O *1	path	絶対パス/相対パス指定	M
<extraction_target>	抽出対象リスト	O *2	path	絶対パス/相対パス指定	M
<extraction_outputdir>	抽出結果ENFの出力フォルダ	O *2	path	絶対パス/相対パス指定 既存のフォルダか、その一階層下のフォルダを指定する	M
<workdir>	ワークフォルダ	M	path	絶対パス/相対パス指定 UNCパスは非サポート	M
<logfile>	ログファイル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

- *1 パート、アセンブリ削除の際は必須
- *2 パート、アセンブリ抽出の際は必須

セクション: INTRCHK

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<csvfile>	結果CSVファイル	M	path	絶対パス/相対パス指定	M
<parameterfile>	パラメータファイル	M	path	絶対パス/相対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス/相対パス指定 UNCパスは非サポート	M
<logfile>	ログファイル	O	path	絶対パス/相対パス指定	M

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: POLYGON

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	M	path	絶対パス/相対パス指定	M
<parameterfile>	パラメータファイル	M	path	絶対パス/相対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス/相対パス指定 UNCパスは非サポート	M
<logfile>	ログファイル	M	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファイル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: PLYGNOPTIMIZE

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	M	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	M	path	絶対パス/相対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス UNCパスは非サポート	M
<parameterfile>	パラメータファイル	M	path	絶対パス/相対パス指定	M
<logfile>	ログファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファイル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

セクション: ENFEDITOR

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<scriptfile>	ENF編集用Rubyスクリプトファイル	M	path	絶対パス指定	M

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<inputfile>	入力ファイル	O *1	path	絶対パス/相対パス指定	M
<outputfile>	出力ファイル	O *2	path	絶対パス/相対パス指定	M
<bomfile>	属性のリストファイル	O *3	path	絶対パス/相対パス指定	M
<mapfile>	マッピングファイル	O *3	path	絶対パス/相対パス指定	M
<namelist>	削除対象リスト	O *4	path	絶対パス/相対パス指定	M
<extraction_target>	抽出対象リスト	O *4	path	絶対パス/相対パス指定	M
<extraction_outputdir>	抽出結果ENFの出力フォルダ	O *4	path	絶対パス/相対パス指定 既存のフォルダか、その一階層下のフォルダを指定する	M
<workdir>	ワークフォルダ	M	path	絶対パス/相対パス指定 UNCパスは非サポート	M
<parameterfile>	パラメータファイル	O	path	絶対パス/相対パス指定	M
<logfile>	ログファイル	O	path	絶対パス/相対パス指定	M
<xmllogfile>	XML形式のログファイル	O	path	絶対パス/相対パス指定	M
<timeout>	タイムアウト時間	O	value	単位は分	M

- *1 scriptfileとして標準搭載スクリプト(アセンブリ編集、属性編集)を指定する場合には必須
- *2 scriptfileとして標準搭載スクリプト(アセンブリ編集)を指定する場合には必須
- *3 scriptfileとして標準搭載スクリプト(属性編集)を指定する場合のみ有効
- *4 scriptfileとして標準搭載スクリプト(アセンブリ編集)を指定する場合のみ有効

セクション: DSCHECK

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<scriptfile>	DFM Studio Checker 用スクリプトファイル	M	path	絶対パス指定	M
<inputfile>	入力ファイル	M	path	絶対パス指定	M
<outputfile>	出力ファイル	M	path	絶対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス指定	M
<parameterfile>	パラメータファイル	O	path	絶対パス指定	M
<logfile>	ログファイル	O	path	絶対パス指定	M

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<xmllogfile>	XML 形式のログファイル	O	path	絶対パス指定	M
<timeout>	タイムアウト時間	O	Value	単位は分	M

セクション: DSREPORT

M:必須/O:任意

タグ	説明	M/O (タグ)	属性	説明	M/O (属性)
<scriptfile>	DFM Studio Reporter用スクリプトファイル	M	path	絶対パス指定	M
<inputfile>	入力ファイル	M	path	絶対パス指定	M
<workdir>	ワークフォルダ	M	path	絶対パス指定	M
<parameterfile>	パラメータファイル	O	path	絶対パス指定	M
<logfile>	ログファイル	O	path	絶対パス指定	M
<xmllogfile>	XML 形式のログファイル	O	path	絶対パス指定	M
<timeout>	タイムアウト時間	O	Value	単位は分	M

2.4.2. 実行例

I. シナリオXML、シナリオログの指定

• シナリオXML の指定

シナリオXMLを利用するためには、コンフィグファイルの共通セクションにシナリオXMLファイルのパスを記述します。

```
SCENARIO_FILE [シナリオXMLファイルパス]
```

もしくはElybatchの引数 -S で、シナリオXMLを指定する事が可能です。コンフィグファイルと引数の両方に指定した場合は、引数で指定されたファイルが優先されます。

• シナリオログファイル

以下のキーワードをコンフィグファイルに設定することで、各コンポーネントの実行結果や実行状態をログに記述することができます。

```
SCENARIO_LOG [LOGファイルパス]
```

Elybatchの引数 -U により、シナリオログファイル名を指定できます。引数とコンフィグファイルの両方に指定されている場合は、引数で指定したファイルが優先されます。

シナリオログの形式

```
<SCENARIO_LOG>
  <PROGRESS> . . . シナリオの実行状態
    <totalstep count=1> . . . シナリオの総ステップ数
    <currentstep count=1> . . . 現在実行中のステップ
  </PROGRESS>
  <DEK> . . . 各セクションの実行情報
    <returncode value="1002"/> . . . リターンコード
    <inputfile path="XXXX"/> . . . 入力ファイル名
    . . . .
  </DEK>
</SCENARIO_LOG>
```

- 連続実行条件

直前に実行したリターンコードを判別し、エラーになった場合は、その時点でシナリオを途中で終了します。(ただし、1000～1100のエラーについては、その後も継続して処理を行います。)

II. 実行例

- シナリオ例 (a)

```
<?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>
```

実行コマンド例

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

- シナリオ例 (b)

```
<?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>
```

実行コマンド例

```
"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"
-w"C:\work" -S"C:\scenario\scenario.xml"
```

2.5. パラメータファイル

パラメータファイルは変換時に使用する変換オプション等を指定するファイルです。
詳細は、パラメータマニュアルを確認してください。

2.6. エラーコード

エラーコードは、実行終了時にワークディレクトリに作成される、"elybatch.rc"というファイルから取得可能です。エラーコードの一覧は"ERRCODE.pdf"をご覧ください。

2.6.1. 注意点

- 強制終了処理

ワークディレクトリに"killprocess"という名前のファイルを作成すると、一連の処理を強制終了します。その際には、エラーコード 999 が返ります。

- 使用禁止文字チェック

出力ファイル(引数"-o"またはシナリオ<outputfile>)として指定したファイルに以下の文字が含まれている場合は、変換を中止します。その際には、エラーコード 33が返ります。
(変換途中で生成されるファイルに関しては、"_" アンダースコアで置き換えます。)

- WINDOWS予約語

最初の「.」までの文字列、もしくは全体文字列が大文字/小文字を問わず以下の文字列になる場合

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

- WINDOWSファイル名使用禁止文字

\/:*?"<>|

3. ASFALIS Component

3.1. Adapter

アダプタはCAD環境を利用するもの(CAD組込)と、CAD環境を必要としないもの(独立型)に分類されます。また、マルチプロセスでの処理に対応した「新アダプタ」と、既存の「旧アダプタ」が存在します。ここでは、各アダプタの設定方法や動作仕様に関して説明します。

	CAD環境	ENF Writer		ENF Reader	
		新アダプタ	旧アダプタ	新アダプタ	旧アダプタ
CATIA V5(CAA)	要	○	◎	◎	○
3DEXPERIENCE	要	◎	-	◎	-
CATIA V4	不要	-	◎	-	◎
NX I-deas	要	-	◎	-	◎
Parasolid	不要	-	◎	-	◎
NX(UFUNC)	要	○	◎	◎	○
Creo Parametric	要	○	◎	◎	○
Autodesk Inventor	要	△	◎	△	◎
Creo Elements/ Direct Modeling	要	-	◎	-	◎
ACIS	不要	-	◎	-	◎
STEP	不要	-	◎	-	◎
STEP AP242 BOM	不要	◎	-	◎	-
IGES	不要	-	◎	-	◎
SOLIDWORKS	要	◎	○	◎	○
JT	不要	◎	○	◎	○
PLM XML	不要	◎	-	◎	-
iCAD	不要	-	◎	-	◎
CATIA V5 (standalone)	不要	-	◎	-	◎
Creo Parametric (standalone)	不要	-	◎	N/A	N/A
NX (standalone)	不要	-	◎	N/A	N/A
CADmeister (standalone)	不要	-	◎	-	◎
XVL	不要	N/A	N/A	-	◎

STL	不要	N/A	N/A	-	◎
3D PDF	不要	N/A	N/A	-	◎
3DXML	不要	-	◎	N/A	N/A

◎：デフォルトで使用するアダプタ

○：使用可能

△：使用可能だがサポート対象外

-：非対応

N/A：対応製品なし

以下、本章内では以下のフォルダに該当するフォルダを<ASFALIS Components>と表記します。

- <ASFALIS Componentパッケージ>\bin

または

- <ASFALIS Componentsインストールフォルダ>\module

3.1.1. CATIA V5 to ENF / ENF to CATIA V5

I. 必要実行モジュール等

- アダプタ
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
- 旧アダプタ
<ASFALIS Components>\CaaExe
<ASFALIS Components>\tools
<ASFALIS Components>\Cat4Exe

II. 設定方法

- CATIA V5 to ENF
 - PATH_FROM_TRANSLATOR
CATIA V5の本体(CNEXT.exe)がインストールされているディレクトリ
(例) C:\Program Files\Dassault Systemes\B24\win_b64\code\bin
 - FROM_TRANSLATOR
CATIA V5旧アダプタ本体があるディレクトリ
(例) C:\elysium\bin\CaaExe\win\B24

- ENF to CATIA V5

- PATH_TO_TRANSLATOR

CATIA V5の本体(CNEXT.exe)がインストールされているディレクトリ

(例) C:\Program Files\Dassault Systemes\B24\win_b64\code\bin

- TO_TRANSLATOR

CATIA V5旧アダプタ本体があるディレクトリ

(例) C:\elysium\bin\CaaExe\win\B24



注意)

PATH_FROM_TRANSLATOR, PATH_TO_TRANSLATORで指示したCATIA V5本体のバージョンと、FROM_TRANSLATOR, TO_TRANSLATORで指示したアダプタのバージョンを揃えてください。

III. 入出力仕様

- CATIA V5 to ENF

入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは、パートファイルの場合はXXX.CATPartを、アセンブリの場合はトップアセンブリのXXX.CATProductを指定します。サブアセンブリおよびパートファイルは、変換環境のCATIA V5で開けるよう配置されている必要があります。

アセンブリ構成要素の検索順はCATIA V5の以下の設定で変更可能です。

- [ツール] > [オプション] > [一般] > [ドキュメント] > 「リンクされたドキュメントを検索」

- ENF to CATIA V5

出力ファイル(引数"-o"またはシナリオ<outputfile>)で拡張子無しのファイル名を指定します。

パートの場合は"XXX.CATPart"、アセンブリの場合はトップアセンブリ名が"XXX.CATProduct"と一連のパートファイルが、ワークディレクトリに出力ファイルとして作成されます。

IV. 注意点

1. CATIA V5はログインユーザごとにDSLS(Dassault Systemes License Server)にてライセンス設定をする必要があります。アダプタを実行するログインユーザでCATIA V5 本体を起動できる環境を作成してください。
2. PATH_FROM_TRANSLATOR, PATH_TO_TRANSLATORで指示されたCATIA V5 本体のバージョンとFROM_TRANSLATOR, TO_TRANSLATORで指示されたアダプタのバージョンをあわせてください。
3. CATIA V5 の導入環境によっては、設定方法に記載したとおりに設定しても、動作しない場合があります。その場合は通常のCATIA V5 使用時の環境ファイルを、コンフィグファイルのキーワードCATIAV5_EnvPath に設定してください。

- (例)

作成した環境ファイル：CATIA.V5R21.B21.txt

環境ファイルを配置したディレクトリ：C:\elysium\CATEnv

設定するコンフィグファイルのキーワード：

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

4. 以下の実行モジュールに対してWindows ファイアウォール経由の通信を許可する必要があります。コントロールパネルから設定を行うか、スクリプト(firewall_add.bat)を管理者権限で実行してください。

- アダプタ

CATIA V5 to ENF実行時 : caa2fbt.exe

ENF to CATIA V5実行時 : fbt2caa.exe

スクリプトは以下のフォルダに存在します。

- 64bit

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

ファイアウォール例外設定を行わずに変換を実行すると、初回のみ、ファイアウォールによるブロックを警告するダイアログが表示されます。「ブロックを解除する」を選択してください。

- 旧アダプタ

CATIA V5 to ENF実行時 : caa2enf.exe

ENF to CATIA V5実行時 : enf2caa.exe

スクリプトは実行モジュールと同じフォルダに存在します。

V. 制限事項

1. マルチバイト文字および以下の文字はファイル名に含めることが出来ない。

\\/:*?"<>|

以下の文字は可。

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

(アダプタの処理)

ENF to CATIA V5 : 使用禁止文字をアンダースコアに変更する。(出力ファイル名に使用禁止文字を指定していた場合は、新アダプタではリネームして正常終了する。旧アダプタではリネームせずエラーを返す。)

CATIA V5 to ENF : エラー (そもそもCATIA V5がそのようなファイル名のデータを作らない)

2. パート名、アセンブリ名及びインスタンス名に以下の文字を含めることが出来ない。

!:

(アダプタの処理)

ENF to CATIA V5 : アンダースコアに変更

3.1.2. 3DEXPERIENCE to ENF / ENF to 3DEXPERIENCE

I. 必要実行モジュール等

- アダプタ

<ASFALIS Components>\CADFeature\Batch\195

<ASFALIS Components>\CADFeature\Batch\3dex.x64

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

<ASFALIS Components>\CADFeature\common

<ASFALIS Components>\tools

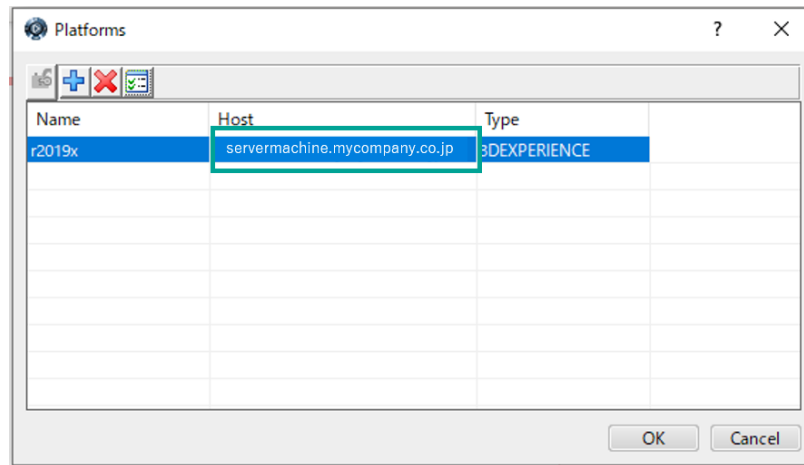
II. 設定方法

- 3DEXPERIENCE to ENF
 - FROM_TRANSLATOR
3DEXPERIENCE アダプタの CAD バージョン名
(例) R2018x
 - FROM_CAD_INSTALL_DIR
3DEXPERIENCE 本体がインストールされているフォルダ
(例) C:\Program Files\Dassault Systemes\B420\win_b64
- ENF to 3DEXPERIENCE
 - TO_TRANSLATOR
3DEXPERIENCE アダプタの CAD バージョン名
(例) R2018x
 - TO_CAD_INSTALL_DIR
3DEXPERIENCE 本体がインストールされているフォルダ
(例) C:\Program Files\Dassault Systemes\B420\win_b64
- 共通
 - 3DEX_Server
3DSpaceのURL
(例) <https://servermachine.mycompany.co.jp:8080/3dspace>
 - 3DEX_LoginTicket
ログインチケット文字列
(例) NEE3QkYxNzY4N0MyNDJDMEFEMzVFNUU5NDE4MEEyNUV8ZGV2MjAxN3h8ZGV2

これらの情報を確認する方法については以下を参照してください。

3DSpace の URL は以下のいずれかの方法で確認することができます。

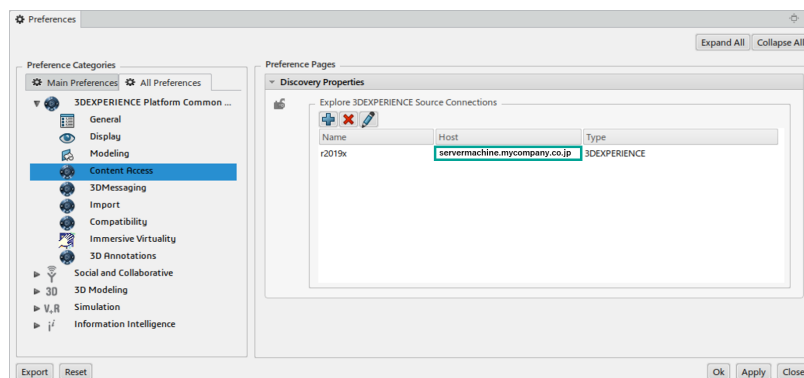
- a. 3DEXPERIENCE を起動する際に表示されるダイアログで確認する



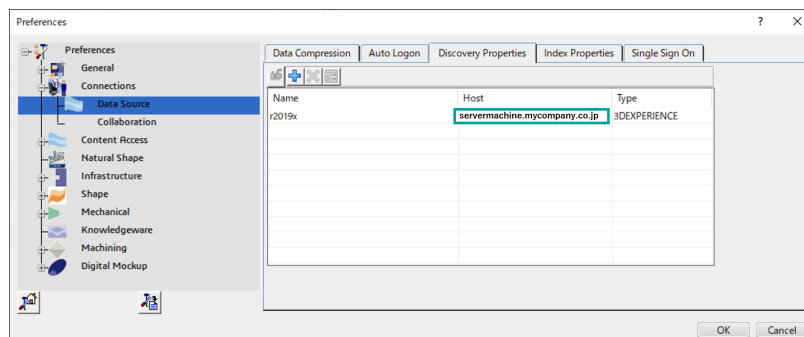
- b. 3DEXPERIENCE の設定画面から確認する

3DEXPERIENCE のメニューから Preference を選択すると表示される設定画面で確認することができます。

R2019x以降



R2018x以前





ログインチケット文字列は以下の手順で取得することができます。

1. 3DEXPERIENCE を起動します。
2. コンパスから "マイ ソーシャルおよびコラボレーション アプリケーション" を選択します。
3. "Collaboration & Approvals" を選択します。
4. 表示されたページで "エクスペリエンスの構成" - "ログインチケットの管理" を選択します。
5. [作成] を選択します。
6. 表示された画面にある "Generated ticket" の値を確認します。

詳細については 3DEXPERIENCE マニュアルを参照してください。

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

III. 入出力仕様

- 3DEXPERIENCE to ENF

入力ファイル (引数"-i"またはシナリオ<inputfile>) として、3DEXPERIENCE のモデル検索条件を記述したテキストファイルを指定します。テキストファイルには以下の内容を記述してください。

- #3DEXPERIENCE (固定キーワード、省略可)
- セクション名[Model]
- 検索属性キー、値

以下の属性キーが使用可能です。

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

使用例:

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

- ENF to 3DEXPERIENCE

出力ファイル(引数"-o"またはシナリオ<outputfile>)でテキストファイルを指定します。
このテキストファイルに、変換で作成される3DEXPERIENCEのモデル検索条件が記述されます。

IV. 制限事項

1. インスタンス名に以下の文字を含めることが出来ない。

!:\|/

(アダプタの処理)

ENF to 3DEXPERIENCE : アンダースコアに変更

3.1.3. CATIA V4 to ENF / ENF to CATIA V4

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\Cat4Exe
<ASFALIS Components>\tools

II. 設定方法

- CATIA V4 to ENF
 - FROM_TRANSLATOR
CATIA V4アダプタ本体のあるフォルダ
(例) C:\elysium\bin\Cat4Exe\win
- ENF to CATIA V4
 - TO_TRANSLATOR
CATIA V4アダプタ本体のあるフォルダ
(例) C:\elysium\bin\Cat4Exe\win

III. 入出力仕様

- CATIA V4 to ENF
引数"-i"またはシナリオ<inputfile>で入力ファイル名"XXX.model"を指定する。
- ENF to CATIA V4
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.model"を指定する。

IV. 注意点

- CATIA V4 to ENFで複数のタイトルが含まれたCATIA V4データを変換するには 複数タイトルの中の1つを選択して、Elybatchに指示することで変換が可能。

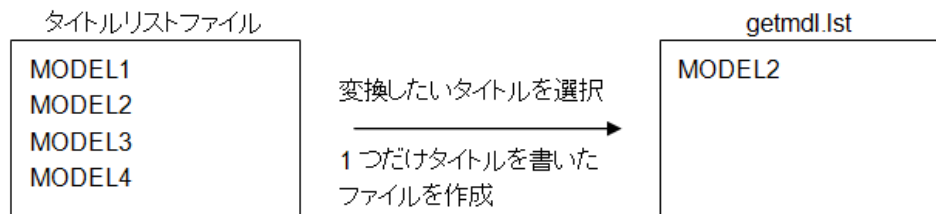
<方法>

以下のモジュールを使ってCATIA V4データから、タイトルのリストを取得する。

<コンポーネントが存在するフォルダ>\bin\Cat4Exe\win\getmdl.exe

```
getmdl.exe -i<CATIAV4 data> -o<title list file>
```

上記のようにコマンドを実行し、タイトルリストファイルを作成する。
タイトルリストファイルにはCATIA V4 データに含まれる全てのモデルタイトルが書かれるので、
変換したいタイトルを選び1つだけタイトルを書いたgetmdl.lstファイルを作成する。



作成したgetmdl.lstをelybatchの引数"-w"またはシナリオ<workdir>で指定したワークディレクトリに置き、変換したいCATIA V4データを変換すると選択したタイトルのモデルのみを変換処理する。

V. 制限事項

1. パート・アセンブリ名にマルチバイト文字を含めることが出来ない。
(アダプタの処理)
ENF to CATIA V4：アンダースコアに変更
2. パート・アセンブリ名に英小文字は使用することが出来ない。
(アダプタの処理)
ENF to CATIA V4：英大文字に変更する
3. パート・アセンブリ名は16文字以内で指定する必要がある。
(アダプタの処理)
ENF to CATIA V4：制限文字数以上はカットする。

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

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\IdeasExe
<ASFALIS Components>\tools

II. 設定方法

- NX I-deas to ENF
 - FROM_CAD_START_CMD
NX I-deas本体の起動ファイル
(例) C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
 - FROM_TRANSLATOR
NX I-deasアダプタ本体があるディレクトリ

(例) C:\elysium\bin\IdeasExe\win\id14

- ENF toNX I-deas
 - TO_CAD_START_CMD
NX I-deas本体の起動バッチファイル
(例) C:\CAD\Siemens\NXI6\bin\I-DEASOGL.cmd
 - TO_TRANSLATOR
NX I-deasアダプタ本体があるディレクトリ
(例) C:\elysium\bin\IdeasExe\win\id14

III. 入出力仕様

- NX I-deas to ENF
引数"-i"またはシナリオ<inputfile>で指定するファイルはXXX.mf1またはXXX.arcです。ただしmf1を指定する場合は同ディレクトリ内にmf2が必要となります。
- ENF to NX I-deas
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.mf1"または"XXX.arc"を指定します。

IV. 注意点

1. NX I-deasアダプタを利用する際にはOrbixが起動している必要があります。
2. カスタマイズされたNX I-deas環境の場合、NX I-deas起動コマンド(IDEASOGL.cmd等)の中で、カレントディレクトリを変更している場合があり、NX I-deasアダプタが利用できないことがあります。NX I-deasを正常にインストールした状態のカスタマイズされていない起動コマンドを利用してください。
3. NX I-deasのメモリ設定の上限値を低く設定している場合、メモリ不足で変換処理が出来ない場合があります。環境変数"IDEAS_PARAM<IDEAS version>"で指示されたNX I-deas用パラメータファイルの設定でメモリ制限を変更できます。
4. 古いバージョンのNX I-deasで作成されたmfファイルをNX I-deas to ENFアダプタで変換する際、変換に失敗することがあります。回避策として、アダプタのバージョンにあわせてNX I-deasで保存しなおしてから、変換してください。
5. [NX I-deas 6以降を使用する場合の注意点] トランスレータが動作するために必要な環境変数がNX I-deasの初期インストール時には設定されておりません。以下の環境変数の設定が必要です。
編集するファイル: <Ideasのインストールフォルダ>\bin\setup_varbs.cmd
 - NX I-deas 6.3の場合: IDEAS_PARAM17=<I-deasパラメータファイルのフォルダパス>
 - NX I-deas 6.4の場合: IDEAS_PARAM18=<I-deasパラメータファイルのフォルダパス>
 - NX I-deas 6.5の場合: IDEAS_PARAM18=<I-deasパラメータファイルのフォルダパス>
 - NX I-deas 6.6の場合: IDEAS_PARAM18=<I-deasパラメータファイルのフォルダパス>
 - NX I-deas 6.7の場合: IDEAS_PARAM18=<I-deasパラメータファイルのフォルダパス>

V. 制限事項

1. モデルファイルのパスの長さが拡張子を含めて80文字を超えてはならない。

2. elybatchのワークディレクトリのパスの長さが65文字を超えてはならない。
3. マルチバイト文字、空白および以下の文字はファイル名の一部に含めることが出来ない。
`\\ / < > | ` $ % ^ ; : " ' * ? . &`

出力ファイル名(引数"-o"またはシナリオ<outputfile>)に上記の文字列が含まれている場合、エラー終了とする。

4. パート名、アセンブリ名が80文字を超えてはならない。
 (アダプタの処理)
 ENF to NX I-deas : 制限文字数以上はカットする。
5. Bin名が80文字を超えてはならない。Bin名が全て数字の場合は10文字を超えてはならない
 (アダプタの処理)
 ENF to NX I-deas : エラー終了とする。

3.1.5. Creo Parametric to ENF / ENF to Creo Parametric

I. 必要実行モジュール等

- アダプタ
 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
- 旧アダプタ
 <ASFALIS Components>\ProeExe
 <ASFALIS Components>\tools

II. 設定方法

- Creo Parametric to ENF
 - FROM_CAD_START_CMD
 Creo Parametric本体の起動ファイル
 (例) C:\Program Files\PTC\Creo 4.0\F000\Parametric\bin\parametric.bat
 - FROM_TRANSLATOR
 Creo Parametric旧アダプタ本体があるディレクトリ
 (例) C:\elysium\bin\ProeExe\win\creo40
 - OPT_FROM_TRANSLATOR
 (例) OPT_FROM_TRANSLATOR pro_wait
- ENF to Creo Parametric
 - TO_CAD_START_CMD
 Creo Parametric本体の起動ファイル
 (例) C:\Program Files\PTC\Creo 4.0\F000\Parametric\bin\parametric.bat

- TO_TRANSLATOR
Pro/E旧アダプタ本体があるディレクトリ
(例) C:\elysium\bin\ProeExe\win\creo40
- OPT_TO_TRANSLATOR
(例) OPT_TO_TRANSLATOR pro_wait

(注意事項)

カスタマイズされたCreo Parametric本体の起動ファイルを指定して新アダプタを使用する際には、以下のファイルの編集が必要となる場合があります。



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

編集例：

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

III. 入出力仕様

- Creo Parametric to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは、パートファイルの場合はXXX.prtを、アセンブリファイルの場合は、トップアセンブリのXXX.asmを指定します。拡張子に".1"などがつく場合は、".1"まで指定してください。アセンブリに参照されるパートは全て入力ファイルとして指定したファイルと同じフォルダに存在する必要があります。
- ENF to Creo Parametric
出力ファイル(引数"-o"またはシナリオ<outputfile>)として拡張子なしのファイル名を指定します。パートの場合はXXX.prt.1、アセンブリの場合は、XXX.asm.1と一連のパートファイルが出力されます。

IV. 注意点

1. config.proの設定は行わないで、Creo Parametricがインストールされた直後の状態で利用することを推奨します。
2. Creo Parametricのコンフィグ設定(config.sup または config.pro)に以下のオプションが設定されている場合に、モデル依存で変換に失敗することがあります。その場合、以下のオプションを no に設定することで、問題を回避できます。
show_geom_checks_on_creation yes
3. Creo Parametricの表示モード(マスタ簡略表示、ジオメトリ表示、グラフィック表示)のうち、グラフィック表示の場合は変換に失敗します。グラフィック表示以外で保存しなおして下さい。
4. 新アダプタにおいてもマルチプロセスでの処理には非対応となります。
5. Creo Parametric 7.0 以降で、Creo Parametric のコンフィグ設定 (config.sup) に以下のオプションが設定されている場合、変換に失敗します。その場合、以下のオプションの値を always に変更

することで、問題を回避することができます。

```
open_protk_signed_apps prompt
または
open_protk_signed_apps never
```

V. 制限事項

1. ファイル名に関する制限

- i. Creo Parametric 3.0以前では、ファイル名は数字、アルファベット、マイナス、アンダースコアのみを含むことができる。

Creo Parametric 4.0以降では、上記に加えてマルチバイト文字も含むことができる。

(アダプタの処理)

ENF to Creo Parametric：使用禁止文字はアンダースコアに変更する。

- ii. ファイル名の1文字目にマイナスを使用することはできない。

2. 拡張子を除いたファイル名の長さは31文字まで。

(アダプタの処理)

ENF to Creo Parametric：制限文字数を超えた場合は32文字目以降をカットする。

出力先に同名のファイルが存在する場合、末尾に枝番をつけた別名のファイルを作成する。

元のファイル名が31以内であっても、追加される枝番と合わせて31文字を超えてしまう場合には元のファイル名の末尾をカットして31文字に収まるようなファイル名となる。

3. ENF to Creo Parametricのテンプレート変換に関する制約事項、注意点

- i. パートのテンプレートファイルがIPS単位系の場合、単位系が正しく変換されません。
- ii. アセンブリとパートで長さの異なる単位系のテンプレートを用いた場合に、アセンブリの配置が不正となります。
- iii. アセンブリモデルの変換時に、アセンブリ/パートどちらかのテンプレートのみを指定している場合、下記の動作となります。
 - パートのテンプレートのみが指定されている場合
 - パート：テンプレートが適用されて変換される
 - アセンブリ：テンプレートが適用されないで変換される
 - アセンブリのテンプレートのみが指定されている場合
 - パート：テンプレートが適用されないで変換される
 - アセンブリ：テンプレートが適用されて変換される
- iv. アセンブリモデルに含まれる、キルト、データムカーブなどの形状要素に関しては、パートのテンプレートが適用されて変換されます。
- v. テンプレートの単位系にユーザ定義単位が使用されている場合は、mmKs座標系（mm kg Second）に変更されます。
- vi. Creo Parametric自体の既知の問題の影響により、単位系が"mmKs単位系"に変わってしまう現象が確認されております。

4. ENF to Creo Parametric非表示ソリッドの変換仕様について

1パート内に複数のソリッドが存在する場合、一部のソリッドのみ非表示にすることはできません

ん。非表示ソリッドの変換仕様は下記のとおりです。

- i. ENF Body内のClosed Volumeに表示、非表示が混在する場合
非表示のClosed Volumeは「PRO_LAYER_FEAT」タイプの非表示レイヤに登録される。
(非表示のClosed Volumeは非表示レイヤに登録されるが、Creo Parametricの動作仕様上、非表示のClosed Volumeも表示される。)
- ii. ENF Body内のClosed Volumeが全て非表示の場合
「PRO_LAYER_SOLID_GEOM」および「PRO_LAYER_FEAT」タイプの非表示レイヤに登録する。(全てのClosed Volumeが非表示として変換される。)

3.1.6. NX to ENF / ENF to NX

I. 必要実行モジュール等

- アダプタ
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
- 旧アダプタ
<ASFALIS Components>\UgExe
<ASFALIS Components>\tools

II. 設定方法

- NX(UFUNC) to ENF
 - FROM_CAD_INSTALL_DIR
NX本体のインストールされたディレクトリ(UGII_BASE_DIR)
(例) C:\Program Files\Siemens\NX 10.0
 - FROM_TRANSLATOR
NX旧アダプタ本体があるディレクトリ
(例) C:\elysium\bin\UgExe\win\ug100
- ENF to NX(UFUNC)
 - TO_CAD_INSTALL_DIR
NX本体のインストールされたディレクトリ(UGII_BASE_DIR)
(例) C:\Program Files\Siemens\NX 10.0
 - TO_TRANSLATOR
NX旧アダプタ本体があるディレクトリ
(例) C:\elysium\bin\UgExe\win\ug100

III. 入出力仕様

- NX(UFUNC) to ENF

入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルはXXX.prtです。パートファイルがサブアセンブリを含む場合は参照されるパートファイルは全てロード可能である必要があります。

- ENF to NX(UFUNC)

出力ファイル(引数"-o"またはシナリオ<outputfile>)として拡張子なしのファイル名を指定します。パートの場合は、XXX.prt、アセンブリの場合は、XXX.prtと一連のパートファイルが作成されます。

IV. 注意点

- NXアダプタを使用するためには以下の環境変数が設定されている必要がある。
NX9.0以降: SPLM_LICENSE_SERVER

V. 制限事項

1. NX9.0以前: マルチバイト文字および以下の文字はファイル名の一部に含めることが出来ない
NX10.0以降: 以下の文字はファイル名の一部に含めることが出来ない
\\<>|`:"*?
(アダプタの処理)
ENF to NX: パート名に制限文字が含まれた場合は、アンダースコアに変更。
(出力ファイル名に使用禁止文字を指定していた場合は、新アダプタではリネームして正常終了する。旧アダプタではリネームせずエラーを返す。)
2. NXの制限でパート名(ファイル名)は拡張子を除き 最大239文字
(アダプタの処理)
ENF to NX: 制限文字数以降はカットする。
3. NXの制限でファイルのフルパス(フォルダ名+パート名+拡張子)は最大255文字
(アダプタの処理)
ENF to NX: 制限文字数以降はカットする。

(線幅マッピングファイルについて)

NX ENF Writer で、NXの線幅指定をENFの線幅にマッピングするためのファイルです。
コンフィグキーワードWIDTH_DEF_FILEまたはパラメータWidthDefFileでファイルのパスを指定します。線幅マッピングファイルはテキストファイル (.txt) に下記内容を記述して作成します。

- <キー>=<値>の形式で記述します。指定可能な値は下記のとおりです。

行	キー	説明
1	NX_WIDTH_ORIGINAL	NXの線幅No Changeに対応します
2	NX_WIDTH_THIN	NXの線幅Thick Widthに対応します
3	NX_WIDTH_NORMAL	NXの線幅Normal Widthに対応します

行	キー	説明
4	NX_WIDTH_THICK	NXの線幅Thick Widthに対応します
5	NX_WIDTH_1_013	NXのピクセル幅1に対応します
6	NX_WIDTH_2_018	NXのピクセル幅2に対応します
7	NX_WIDTH_3_025	NXのピクセル幅3に対応します
8	NX_WIDTH_4_035	NXのピクセル幅4に対応します
9	NX_WIDTH_5_050	NXのピクセル幅5に対応します
10	NX_WIDTH_6_070	NXのピクセル幅6に対応します
11	NX_WIDTH_7_100	NXのピクセル幅7に対応します
12	NX_WIDTH_8_140	NXのピクセル幅8に対応します
13	NX_WIDTH_9_200	NXのピクセル幅9に対応します

指定可能な値

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

- キーの記述順・フォーマットが正しくない場合は、読み込みエラーとなり、デフォルトの線幅が使用されます。
- NX_WIDTH_THICK までの4行とNX_WIDTH_1_013以降の9行も含めた13行が記述されている必要があります。
- NX9では線幅0.13mm～2.00mmの9種の各線幅はそれぞれピクセル幅1～9に対応しています。NX10以降、各線幅のピクセル幅は "メニュー > 設定 > ビジュアライゼーション > 線タブ" で設定されています。
- 記述例:1 (デフォルト設定)

```

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

```

- 記述例:2

ASFALIS EX3.1以前のパラメータAdjustWidthForXVL=1に相当

```

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. 必要実行モジュール等

- アダプタ
 <ASFALIS Components>\ParasolExe
 <ASFALIS Components>\tools

II. 設定方法

- Parasolid to ENF
 - FROM_TRANSLATOR
 Parasolidアダプタ本体があるディレクトリ
 (例) C:\elysium\bin\ParasolExe\win

- PARASOLID_SCHEMA
schemaファイルのディレクトリ
- ENF to Parasolid
 - TO_TRANSLATOR
Parasolidアダプタ本体があるディレクトリ
(例) C:\elysium\bin\ParasolExe\win
 - PARASOLID_SCHEMA
schemaファイルのディレクトリ

III. 入出力仕様

- Parasolid to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルはXXX.x_t, XXX.x_b, XXX.xmt_txt, XXX.xmt_binです。
- ENF to Parasolid
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.x_t", "XXX.x_b", "XXX,xmt_txt", "XXX.xmt_bin"を指定します。

IV. 制限事項

- パート名・アセンブリ名にマルチバイト文字および以下の文字を含むことが出来ない。
: ; * ? " < > | / \
(アダプタの処理)
ENF to Parasolid：アンダースコアに変更

3.1.8. SOLIDWORKS to ENF / ENF to SOLIDWORKS

I. 必要実行モジュール等

- アダプタ
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
- 旧アダプタ
<ASFALIS Components>\SwExe
<ASFALIS Components>\tools

II. 設定方法

- SOLIDWORKS to ENF
 - FROM_TRANSLATOR

SOLIDWORKS旧アダプタ本体のあるフォルダ
(例) C:\elysium\bin\SwExe\win\2013

- ENF to SOLIDWORKS
 - TO_TRANSLATOR
SOLIDWORKS旧アダプタ本体のあるフォルダ
(例) C:\elysium\bin\SwExe\win\2013

III. 入出力仕様

- SOLIDWORKS to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルはパートファイルの場合は、XXX.sldprtを、アセンブリファイルの場合は、トップアセンブリのXXX.sldasmを指定します。
- ENF to SOLIDWORKS
出力ファイル(引数"-o"またはシナリオ<outputfile>)として拡張子なしのファイル名を指定します。パートファイルの場合は、XXX.sldprtを、アセンブリファイルの場合は、トップアセンブリのXXX.sldasmと一連のパートファイルが作成されます。

IV. 注意点

1. SOLIDWORKS本体(sldworks.exe)のプロセスが存在していると、変換できないため、同一マシンで同時に2つの変換を行うことが出来ません。また、新アダプタにおいてもマルチプロセスでの処理には非対応となります。
2. 旧アダプタでは複数のバージョンのSOLIDWORKSを使い分けることはできません。複数のバージョンがインストールされている場合は、最後に起動したバージョンのSOLIDWORKSが変換に使用されます。
3. PMI を変換するためには、新アダプタを利用する必要があります。旧アダプタで PMI を変換することはできません。
4. 変換前に、SOLIDWORKS上でテンプレートをあらかじめ指定しておく必要があります。
(正しくテンプレートが指定されていない場合、SOLIDWORKS上で新規にパーツまたはアセンブリを作成しようとする、「テンプレートが見つかりません」というエラーメッセージが表示されます)
5. 旧アダプタ使用時、OSがWindows 8.1およびWindows 10で、UACが有効になっている場合、下記の設定が必要となります。
 - i. 一時的にUACを無効に設定し、マシンを再起動してください
 - ii. マクロ (load_addin.swp) を編集します。
SOLIDWORKSを管理者として起動します。メニューから [ツール] > [マクロ] > [編集] を選択し、load_addin.swpを開いてください。load_addin.swpはアダプタと同じフォルダに存在します。
(例) D:\bin\SwExe\win\2015\load_addin.swp
下記の行をdllのフルパスに変更し、マクロを保存してください。
 - (変更前)


```
Const sAddinName As String = "sw2nut.dll"
```

- (変更後 例)

```
Const sAddinName As String = "D:\bin\SwExe\win\2015\sw2nut.dll"
```

- iii. マクロを実行します。
SOLIDWORKSのメニューから [ツール] > [マクロ] > [実行] を選択し、ii.で保存したマクロを実行してください。
- iv. SOLIDWORKSのメニューから [ツール] > [アドイン] を選択し、表示されるアドインウィンドウに、「Sw2nutAddin」と表示されていることをご確認ください
- v. UACを有効に戻し、マシンを再起動してください

3.1.9. Inventor to ENF / ENF to Inventor

I. 必要実行モジュール等

- アダプタ
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
- 旧アダプタ
<ASFALIS Components>\AiExe
<ASFALIS Components>\tools

II. 設定方法

- Inventor to ENF
 - FROM_TRANSLATOR
Inventor|旧アダプタ本体のあるフォルダ
(例) C:\elysium\bin\AiExe\win\2014
- ENF to Inventor
 - TO_TRANSLATOR
Inventor|旧アダプタ本体のあるフォルダ
(例) C:\elysium\bin\AiExe\win\

III. 入出力仕様

- Inventor to ENF

入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルはパートファイルの場合はXXX.iptを、アセンブリファイルの場合はトップアセンブリのXXX.iamを指定します。

- ENF to Inventor

出力ファイル(引数"-o"またはシナリオ<outputfile>)として拡張子なしのファイル名を指定します。変換されるデータがパートの場合はXXX.iptが、アセンブリの場合はXXX.iamと一連のパートファイルが作成されます。

IV. 注意点

1. 同一マシンに複数のバージョンのInventorをインストールしてバージョンを使い分けてアダプタを使用することはできません。複数のバージョンがインストールされている場合は、最後に起動したバージョンのInventorが変換に使用されます。
2. 新アダプタを使用するには、変換前にモジュールを登録する必要があります。使用するInventorのバージョンに合わせて、インストールフォルダにある下記バッチファイルを「管理者として実行」してください。

64bit

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

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

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\OsdExe
<ASFALIS Components>\tools

II. 設定方法

- Creo Elements/Direct to ENF
 - FROM_CAD_START_CMD
Creo Elements/Direct本体の起動実行形式
(例) C:\PTC\Creo Elements\Direct Modeling 18.1\binNT\SolidDesigner.exe
 - FROM_TRANSLATOR
Creo Elements/Directアダプタ本体のあるフォルダ
(例) C:\elysium\bin\OsdExe\win\CM18
- ENF to Creo Elements/Direct
 - TO_CAD_START_CMD
Creo Elements/Direct本体の起動実行形式
(例) C:\PTC\Creo Elements\Direct Modeling 18.1\binNT\SolidDesigner.exe
 - TO_TRANSLATOR
Creo Elements/Directアダプタ本体のあるフォルダ
(例) C:\elysium\bin\OsdExe\win\CM18

III. 入出力仕様

- Creo Elements/Direct to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは XXX.pkg, XXX.sda, XXX.sdp, XXX.sd です。
- ENF to Creo Elements/Direct
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.pkg"を指定します。

IV. 注意点

1. Creo Elements/Direct to ENFで生成されたENFをCADdoctorで読み込むにはコンフィグファイルに
※MODIFY_ENF_HEADER 112001
を記述しておく必要がある。

V. 制限事項

1. 以下の文字をファイル名の一部に含めることが出来ない。
\\ / < > | : ; , " ' ? *
2. 以下の文字はパート・アセンブリ名に含めることが出来ない。
\\ / < > [] | : ; , " ' ~ ^ * + = ? ! @ # \$ % & 空白
(アダプタの処理)
ENF to Creo Elements/Direct：アンダースコアに変更
3. パート名・アセンブリ名は80文字を超えてはならない。
(アダプタの処理)
ENF to Creo Elements/Direct：制限文字数以降はカット
4. アセンブリ名、パート名、インスタンス名、座標系名、注記に日本語が含まれている場合の制限
Creo Elements/Direct to ENFでパラメータWriteOldFileに2または3を指定した場合、日本語文字変換はサポートされません。0を指定した場合や指定なしの場合は日本語文字変換がサポートされます。

3.1.11. ACIS to ENF / ENF to ACIS

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\AcisExe
<ASFALIS Components>\tools

II. 設定方法

- ACIS to ENF
 - FROM_TRANSLATOR
ACISアダプタ本体のあるフォルダ
(例) C:\elysium\bin\AcisExe\win

- ENF to ACIS
 - TO_TRANSLATOR
ACISアダプタ本体のあるフォルダ
(例) C:\elysium\bin\AcisExe\win

III. 入出力仕様

- ACIS to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは XXX.sat です。
- ENF to ACIS
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.sat"を指定します。

3.1.12. IGES to ENF / ENF to IGES

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\IgesExe
<ASFALIS Components>\tools

II. 設定方法

- IGES to ENF
 - FROM_TRANSLATOR
IGESアダプタ本体のあるフォルダ
(例) C:\elysium\bin\IgesExe\win
- ENF to IGES
 - TO_TRANSLATOR
IGESアダプタ本体のあるフォルダ
(例) C:\elysium\bin\IgesExe\win

III. 入出力仕様

- IGES to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは XXX.igs, "XXX.iges"です。
- ENF to IGES
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.igs", "XXX.iges"を指定します。

IV. 注意点

1. 使用禁止文字処理は行わない。CADの入出力に依存する。

3.1.13. STEP to ENF / ENF to STEP

I. 必要実行モジュール等

- アダプタ
 <ASFALIS Components>\StepExe
 <ASFALIS Components>\tools

II. 設定方法

- STEP to ENF
 - FROM_TRANSLATOR
STEP to ENFアダプタ本体のあるフォルダ
(例) C:\elysium\bin\StepExe\win
- ENF to STEP
 - TO_TRANSLATOR
ENF to STEPアダプタ本体のあるフォルダ
(例) C:\elysium\bin\StepExe\win

III. 入出力仕様

- STEP to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは "XXX.stp", "XXX.step" です。
- ENF to STEP
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.stp", "XXX.step", "XXX.stpZ"を指定します。

IV. 注意点

1. ENF to STEPのプロトコルおよびファイルバージョンに関する注意点 (1)
フォーマット側の制限のため、以下の要素、属性および機能はAP203のed2またはAP214, AP242のSTEPを出力する場合のみ変換対象となります。
 - 色
 - 線種・線幅
 - 点種
 - レイヤ
 - 表示・非表示
 - 座標系
 - データム平面
 - データム軸
 - 複数ファイル形式(外部パート参照複数ファイル形式のみ対応)
2. ENF to STEPのプロトコルおよびファイルバージョンに関する注意点 (2)
以下の要素、属性および機能はAP242のSTEPを出力する場合のみ変換対象となります。
 - PMI

- ユーザ属性
- 検証プロパティ

V. 制限事項

1. 以下の文字をファイル名の一部に含めることが出来ない。
`\ / < > | : ; , " ? *`
2. パート名・アセンブリ名は80文字を超えてはならない。
 (アダプタの処理)
 ENF to STEP：制限文字数以降はカット
3. STEPの仕様ではファイル名に非ASCII文字を含めてはならない。
 (アダプタの処理)
 ENF to STEP：複数ファイル形式のSTEPへ変換する場合、通常はパート名をファイル名として使用する。パート名に非ASCII文字が含まれるモデルを複数ファイル形式のSTEPへ変換する場合は、ASCII文字に置き換えた文字列をファイル名とする。
 単一ファイルのSTEPを出力する場合で、出力ファイル名に非ASCII文字を指定した場合は、文字列の置き換えは行わない。

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

I. 必要実行モジュール等

- アダプタ
 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

IT形式の外部参照パート変換オプションの利用時は追加で下記フォルダも必要

- アダプタ
 64bit
 <ASFALIS Components>\CADFeature\Batch\184
 <ASFALIS Components>\CADFeature\Batch\JT.x64

あるいは

- 旧アダプタ
 <ASFALIS Components>\JtExe

STEP形式の外部参照パート変換オプションの利用時は追加で下記フォルダも必要

- アダプタ
 <ASFALIS Components>\StepExe

II. 設定方法

- STEP AP242 BOM to ENF
 - FROM_TRANSLATOR
新アダプタのみのため不要
- ENF to STEP AP242 BOM
 - TO_TRANSLATOR
新アダプタのみのため不要

III. 入出力仕様

- STEP AP242 BOM to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは XXX.stpxあるいはXXX.stpxZです。
外部参照する JTファイルやSTEPファイルを変換する場合は、stpx/stpxZ内に記述されたパスに配置、もしくは stpx/stpxZと同じフォルダに配置します。
- ENF to STEP AP242 BOM
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.stpx"あるいは"XXX.stpxZ"を指定します。
パート形状を JTファイルやSTEPファイルに変換する場合は、XXX.stpx/XXX.stpxZが出力されるフォルダに自動的に出力されます。

IV. 注意点

1. JT形式の外部参照パート変換オプションの利用時は、JTコンポーネントの設定やライセンスも必要となります。また、以下のパラメータ設定も必要となります。

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

2. STEP形式の外部参照パート変換オプションの利用時は、STEPコンポーネントの設定も必要となります。また、以下のパラメータ設定も必要となります。

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

3. STEP AP242 BOMではDEK、JTとSTEPのパラメータを設定可能です。設定時にはマニュアル編集で記述する必要があり、以下のように記述します。

(例)

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

CANVASスクリプトの一般的な編集方法については、CANVASスクリプト編集マニュアルを別途参照してください。

4. STEP AP242 BOM使用時はSTEPアダプタのライセンスを使用します。

V. 制限事項

1. (STEP AP242 BOM to ENF) 外部参照ファイルの形式は JT、STEP以外は非サポートです。また、JTとSTEPを混在させた変換も非サポートです。
2. (ENF to STEP AP242 BOM) 外部参照ファイルの形式は JT、STEP以外は非サポートです。
3. (ENF to STEP AP242 BOM) パート形状を JTファイルに変換する場合は、ファイル形式としてFull Shatter以外の形式は非サポートです。その他 JTファイル変換の制限事項はENF to JTの節 (3.1.16, “PLM XML to ENF / ENF to PLM XML”, V. 制限事項)を参照してください。
4. (ENF to STEP AP242 BOM) 変換対象のアセンブリ内に同名のアセンブリやパートがある場合、自動的にリネームされます。

3.1.15. JT to ENF / ENF to JT

I. 必要実行モジュール等

- アダプタ
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
- 旧アダプタ
<ASFALIS Components>\JtExe
<ASFALIS Components>\tools

II. 設定方法

- JT to ENF
 - FROM_TRANSLATOR
JT 旧アダプタ本体のあるフォルダ
(例) C:\elysium\bin\JtExe\win
- ENF to JT
 - TO_TRANSLATOR
JT 旧アダプタ本体のあるフォルダ
(例) C:\elysium\bin\JtExe\win

III. 入出力仕様

- JT to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは XXX.jt です。
- ENF to JT
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.jt"を指定します。

IV. 注意点

1. ENF to JTのファイルバージョンに関する注意点
JTのフォーマットとしてPMIはファイルバージョン9.4以上でのサポートとなっています。
PMIを変換するパラメータが設定されており、ファイルバージョンの指定が9.4未満の場合は、自動的に9.4に変更して変換されます。

V. 制限事項

1. 拡張子を除いたファイル名の長さは128文字まで。
(アダプタの処理)
ENF to JT：128文字に省略する
2. パート名・アセンブリ名は128文字を超えてはならない。
(アダプタの処理)
ENF to JT：制限文字数以降はカットする
3. デフォルトでは以下の文字をファイル名の一部に含めることが出来ない。
マルチバイト文字および「`」

パラメータ「ReplaceNXProhibitChar」の指定により動作変更が可能

ReplaceNXProhibitChar=0: 上記を制限文字として扱わない

ReplaceNXProhibitChar=1: 上記を制限文字として扱う (デフォルト)

ReplaceNXProhibitChar=2: 「`」のみを制限文字として扱う

(アダプタの処理)

ENF to JT：パート名に制限文字が含まれた場合はアンダースコアに変更

(出力ファイル名に使用禁止文字を指定していた場合は、新アダプタではリネームして正常終了する。旧アダプタではリネームせずエラーを返す。)

3.1.16. PLM XML to ENF / ENF to PLM XML

I. 必要実行モジュール等

- アダプタ
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

IT形式の外部参照パート変換オプションの利用時は追加で下記フォルダも必要

- アダプタ
64bit
<ASFALIS Components>\CADFeature\Batch\184
<ASFALIS Components>\CADFeature\Batch\JT.x64

あるいは

- 旧アダプタ
<ASFALIS Components>\JtExe

NX形式の外部参照パート変換オプションの利用時は追加で下記フォルダも必要

- アダプタ
64bit
<ASFALIS Components>\CADFeature\NX.x64
<ASFALIS Components>\CADFeature\Batch\114
<ASFALIS Components>\CADFeature\Batch\NX.x64

あるいは

- 旧アダプタ
<ASFALIS Components>\UgExe

II. 設定方法

- PLM XML to ENF
 - FROM_TRANSLATOR
新アダプタのみのため不要
- ENF to PLM XML
 - TO_TRANSLATOR
新アダプタのみのため不要

III. 入出力仕様

- PLM XML to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは XXX.plmxml です。
外部参照するJT/NXファイルを変換する場合は、plmxml内に記述されたパスに配置、もしくは
plmxmlと同じフォルダに配置します。
- ENF to PLM XML
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.plmxml"を指定します。
パート形状をJTファイルに変換する場合は、XXX.plmxmlが出力されるフォルダに自動的に出力されます。

IV. 注意点

1. JT/NX形式の外部参照パート変換オプションの利用時は、JT/NXコンポーネントの設定やライセンスも必要となります。
2. PLM XMLではDEK、JTとNXのパラメータを設定可能です。設定時にはマニュアル編集で記述する必要があります、以下のように記述します。
(例)

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

CANVASスクリプトの一般的な編集方法については、CANVASスクリプト編集マニュアルを別途参照してください。

V. 制限事項

1. (PLM XML to ENF) 外部参照ファイルの形式は JT/NX以外は非サポートです。
2. (ENF to PLM XML) 外部参照ファイルの形式は JT以外は非サポートです。
3. (ENF to PLM XML) パート形状を JTファイルに変換する場合は、ファイル形式としてFull Shatter以外の形式は非サポートです。その他 JTファイル変換の制限事項はENF to JTの節を参照してください。

3.1.17. iCAD to ENF / ENF to iCAD

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\IcadExe
<ASFALIS Components>\tools

II. 設定方法

- iCAD to ENF
 - FROM_TRANSLATOR
iCADアダプタ本体のあるフォルダ
(例) C:\elysium\bin\IcadExe\win
- ENF to iCAD
 - TO_TRANSLATOR
iCADアダプタ本体のあるフォルダ
(例) C:\elysium\bin\IcadExe\win

III. 入出力仕様

- iCAD to ENF
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは XXX.icd です。

- ENF to iCAD
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.icd"を指定します。

IV. 制限事項

1. IcadExeをUNCパスにインストールして変換を実行することはできません。
2. インストールフォルダIcadExeには書き込み権限が必要です。
3. (iCAD to ENF) 変換時にパート/アセンブリの共有が解除されます。
4. (iCAD to ENF) 半透明表示要素は変換されません。
5. (ENF to iCAD) 変換時にパート/アセンブリの共有が解除されます。
6. (ENF to iCAD) 同名のパート/アセンブリはリネームされます。
(例) Assy、Assy → Assy_Asm_1、Assy_Asm_2
(例) Body、Body → Body_Body_1、Body_Body_2
7. (ENF to iCAD) パート・アセンブリ名に以下の文字を含めることはできません。
" | \ * ; : < > ? /
(アダプタの処理)
ENF to iCAD：アンダースコアに変更
8. (ENF to iCAD) パート・アセンブリ名は40バイト以内で指定する必要があります。
(アダプタの処理)
ENF to iCAD：制限バイト数以上はカットする。
9. (ENF to iCAD) ファイル名(拡張子は除く)に空白や以下の文字を含めることはできません。
! " \$ ' * , / : ; < = > ? \ ^ ` | ~
(アダプタの処理)
ENF to iCAD：アンダースコアに変更
10. (ENF to iCAD) ファイル名(拡張子は除く)は40バイト以内で指定する必要があります。
(アダプタの処理)
ENF to iCAD：制限バイト数以上はカットする。

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

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\CfioExe
<ASFALIS Components>\tools
<CRESTAMパッケージ>\CRESTAM
 - CRESTAMは、CADmeister (standalone)モジュールを実行するのに必要なDLLなどを含むディレクトリです。

II. 設定方法

- CADmeister to ENF (standalone)
 - FROM_TRANSLATOR

CADmeister (standalone)アダプタ本体のあるフォルダ
(例) C:\elysium\bin\CfioExe\win

- ENF to CADmeister (standalone)
 - TO_TRANSLATOR
 CADmeister (standalone)アダプタ本体のあるフォルダ
(例) C:\elysium\bin\CfioExe\win

III. 入出力仕様

- CADmeister to ENF (standalone)
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは"XXX.cfio"です。
- ENF to CADmeister (standalone)
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.cfio"を指定します。

IV. 注意点

1. CADmeister (standalone)アダプタ本体のあるドライブ直下に、CRESTAMを別途配置する必要があります。CRESTAMは日本語環境、英語環境で共通です。
2. 作業フォルダ、CADmeister (standalone)モジュールとCRESTAMは、全て同じドライブにある必要があります。
3. CADmeister自体をインストールしている環境では、CADmeister本体のCRESTAMとは別のドライブへ、ASFALISのアダプタおよびCRESTAMを配置してください。
4. 属性変換を行うためには、以下のファイルの準備および配置が必要になります。
 - i. お客様環境で個別にカスタマイズしたユーザ属性の設定ファイル(bhn.std, bhn.usr 等)をご用意ください。
 - 設定ファイルは以下のフォルダに配置されています。
<CADmeister本体のインストールドライブ>\CRESTAM\KPKG\KPG\Dic
 - CADmeister to ENFの場合は、入力CFIOファイルが作成されたCADmeister環境のファイル、ENF to CADmeisterの場合は、出力CFIOファイルを読むCADmeister環境のファイルが必要です。
 - ii. 個別にカスタマイズしたユーザ属性の設定ファイルの内容を元に、属性変換用のXMLファイル(attr_cfio2enf.xml, attr_enf2cfio.xml)を作成します。i.のDicフォルダ全体を添えてエリジオンのサポート窓口へお問い合わせください。
 - iii. CADmeister環境の設定ファイルをCADmeister (standalone)アダプタ用CRESTAMフォルダ以下にコピーし、置き換えます。
<CADmeister本体のインストールドライブ>\CRESTAM\KPKG\KPG\Dic\bhn.std
<CADmeister本体のインストールドライブ>\CRESTAM\KPKG\KPG\Dic\bhn.usr 等
↓ コピー
<CRESTAM>\KPKG\KPG\Dic\bhn.std
<CRESTAM>\KPKG\KPG\Dic\bhn.usr 等
 - iv. CADmeister (standalone)アダプタ本体のあるフォルダに、ii.で作成された属性変換用のXMLファイルを配置します。

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

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

5. CADmeister (standalone)アダプタは、現状では日本語OSかつ日本語ロケール、あるいは英語OSをサポート対象としております。
6. 出力されたCFIOファイルを閲覧する環境のカラーテーブルをカスタマイズしている場合には、変換を行う環境のカラーテーブルも同じように変更しておく必要があります。変換環境の“<ドライブ>\CRESTAM\CUSTOM\SRC\environ.color”を、出力されたCFIOデータを使用するCADmeisterのカラーテーブルと同じものに差し替えてください。

V. 制限事項

1. コントロールコード、空白および以下の文字はオブジェクト名に含めることが出来ない。
!"\$*,?@\
- (アダプタの処理)
ENF to CADmeister(standalone): アンダースコアに変更する。
2. オブジェクト名は、63バイトを超えてはならない。
(アダプタの処理)
ENF to CADmeister(standalone): 制限文字数以降はカットする。
3. オブジェクト名が重複してはならない。
(アダプタの処理)
ENF to CADmeister(standalone): 末尾に追番を付ける。

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

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\Dsav5Exe
<ASFALIS Components>\tools

II. 設定方法

- CATIAV5 to ENF (standalone)
 - FROM_TRANSLATOR
CATIA V5 (standalone)アダプタ本体のあるフォルダ
(例) C:\elysium\bin\Dsav5Exe\win
- ENF to CATIAV5 (standalone)
 - TO_TRANSLATOR
CATIA V5 (standalone)アダプタ本体のあるフォルダ
(例) C:\elysium\bin\Dsav5Exe\win

III. 入出力仕様

- CATIAV5 to ENF (standalone)
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは、パートファイルの

場合はXXX.CATPartを、アセンブリの場合は、トップアセンブリのXXX.CATProductを指定します。

- ENF to CATIAV5 (standalone)

出力ファイル(引数"-o"またはシナリオ<outputfile>)で拡張子無しのファイル名を指定します。

パートの場合は"XXX.CATPart"、アセンブリの場合はトップアセンブリ名が"XXX.CATProduct"と一連のパートファイルが、ワークディレクトリに出力ファイルとして作成されます。

IV. 注意点

1. 複数同時並行でENF to CATIAV5 (standalone)を実行した場合に正しく変換されない場合がある。

V. 制限事項

1. 以下の文字はファイル名に含めることが出来ない。
\\/:*?"<>|
2. 2バイト目が0x5cとなる文字を含む出力ファイル名の場合、ファイル名の該当箇所が「・」に置換される。
3. 2バイト目が0x5cとなる文字を、出力ファイル(引数"-o"またはシナリオ<outputfile>)で指定するパスのフォルダ名に含めることが出来ない。

3.1.20. Creo Parametric to ENF (standalone)

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\DsaproExe
<ASFALIS Components>\tools

II. 設定方法

- Creo Parametric to ENF (standalone)
 - FROM_TRANSLATOR
Creo Parametric (standalone)アダプタ本体のあるフォルダ
(例) C:\elysium\bin\DsaproExe\win

III. 入出力仕様

- Creo Parametric to ENF (standalone)
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルは、パートファイルの場合はXXX.prtを、アセンブリファイルの場合は、トップアセンブリのXXX.asmを指定します。拡張子に".1"などがつく場合は、".1"まで指定してください。アセンブリに参照されるパートは全て入力ファイルとして指定したファイルと同じフォルダに存在する必要があります。

3.1.21. NX to ENF (standalone)

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\DsanxExe
<ASFALIS Components>\tools

II. 設定方法

- NX to ENF (standalone)
 - FROM_TRANSLATOR
NX (standalone)アダプタ本体のあるフォルダ
(例) C:\elysium\bin\DsanxExe\win

III. 入出力仕様

- NX to ENF (standalone)
入力ファイル(引数"-i"またはシナリオ<inputfile>)として指定するファイルはXXX.prtです。パートファイルがサブアセンブリを含む場合は参照されるパートファイルは全てロード可能である必要があります。

3.1.22. ENF to STL

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\StlExe
<ASFALIS Components>\tools

II. 設定方法

- ENF to STL
 - TO_TRANSLATOR
STLアダプタ本体のあるフォルダ
(例) C:\elysium\bin\StlExe\win

III. 入出力仕様

- ENF to STL
引数"-o"またはシナリオ<outputfile>で出力ファイル名"XXX.stl"を指定します。

3.1.23. ENF to XVL

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\XvlExe
<ASFALIS Components>\tools

II. 設定方法

- ENF to XVL
 - TO_TRANSLATOR
XVLアダプタ本体のあるフォルダ
(例) C:\elysium\bin\XvlExe\win

III. 入出力仕様

- ENF to XVL
出力ファイル(引数"-o"またはシナリオ<outputfile>)として拡張子なしのファイル名を指定します。P-XVLで出力する場合は、"XXX.xv3"が、V-XVL、U-XVLで出力する場合は、"XXX.xv2"が作成されます。複数ファイル書き出しの場合は、"XXX.xv0" が作成されます。

入力ファイルとして、ヒーリング前のENFを指定します。ENF to XVLはヒーリング前のENFを読み込み、ENF to XVLの処理中にヒーリングを実行します。

ElybatchからXVL向けのヒーリングの実行を指示しても、adeKernel.exeでのヒーリングはスキップされます。

V. 制限事項

1. グループ名（アセンブリ名）に以下の文字を使用できない。
" 先頭のアンダースコア 末尾の%N
(アダプタの処理)
"はアンダースコア、先頭のアンダースコアはハイフン、末尾の%Nは#N、にそれぞれ置換する。

3.1.24. ENF to 3D PDF

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\tools

II. 設定方法

- ENF to 3D PDF
設定の必要なし (<ASFALIS Components>\tools の下に全てのモジュールがある状態で固定されており、TO_TRANSLATOR も指定する必要はありません。)

III. 入出力仕様

- ENF to 3D PDF
シナリオ<outputfile>として出力ファイル名 (XXX.pdf) を指定します。
シナリオの設定方法については「[2.4, “XMLシナリオ”](#)」を参照してください。

IV. 制限事項

1. コマンドライン引数による実行には対応していない。

*ENF to 3D PDFと併せて PDF Editor を利用する場合には[3.3.9, “3D PDF Editor”](#)もご参照ください。

3.1.25. 3DXML to ENF

I. 必要実行モジュール等

- アダプタ
<ASFALIS Components>\Dsa3dxmlExe
<ASFALIS Components>\tools

II. 設定方法

- 3DXML to ENF
 - FROM_TRANSLATOR
3DXML アダプタ本体のあるフォルダ
(例) C:\elysium\bin\Dsa3dxmlExe\win

III. 入出力仕様

- 3DXML to ENF
入力ファイル (引数"-i"またはシナリオ<inputfile>) として指定するファイルは XXX.3dxml です。

3.2. Healer, etc.

3.2.1. TOGO / CADCEUS Healer

TOGO, CADCEUS向けのヒーリングを行うコンポーネントです。

I. 必要実行モジュール等

- <ASFALIS Components>\tools

II. 設定方法

- PRODUCT
ENF to CAD (TOGO,CADCEUS用armo)へのヒーリングを実行する場合、100231または100211 をそれぞれ指定。
CAD(TOGO,CADCEUS用 armo) to CADのヒーリングでは123061等と設定。
- ADEK_DIR
ヒーリングプログラムが存在するディレクトリ
(例) \$ESERVER\tools\win
- MODIFY_ENF_HEADER
TOGO, CADCEUS用にヘッダーを書き換える。
103211(NX I-deas to CADCEUS), 104231(Creo Parametric to TOGO)など、CAD側のトランスレータにあわせて設定。

3.2.2. ENF Polygon

指定されたENFファイルの要素をポリゴン化して、ポリゴンを含むENFファイルを出力します。

I. 必要実行モジュール等

- <ASFALIS Components>\tools

II. ENF Polygonの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

III. 設定方法

パラメータファイルでポリゴンの精度、出力形式等を設定します。詳細は、パラメータマニュアルを参照してください。

3.3. Optimizer

3.3.1. PDQChecker

I. PDQ Checkerの実行

1. Elybatchから実行させる場合
下記の二つの方法があります。
 - a. シナリオで実行する (推奨)
「[2.4, “XMLシナリオ”](#)」を参照してください。
 - b. 引数から実行する
elybatch.exe の引数で、-n を指定します。
2. 単独で実行させる場合
以下の引数で"checknut.exe"を実行します。

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

-i	検証する対象のENFファイル
-o	チェック結果のCSVファイル
-e	チェック結果を含んだENFファイル。このENFファイルをModel Viewer, CADdoctorで開くと、チェック結果を確認することができます。
-c	パラメータファイル
-l	ログファイル
-f	フォーマットファイル <コンポーネントが存在するフォルダ>\bin\tools\win\DekCheck.msg

-X	XMLログファイル
----	-----------

実行前に以下の環境変数で、ライセンスを設定する必要があります。

```
ELY_SEC_SERVER = <ライセンスサーバ名>
ELY_SEC_PORT = <ポート番号>
```

戻り値

0	正常終了
1	引数が不足
4	引数が不正
5	パラメータファイル読み込みエラー
6	入力ENFが指定されていない
7	結果出力ファイルが指定されていない
8	フォーマットファイルが指定されていない
9	フォーマットファイルが開けない、または不正
10	ログファイルが開けない
11	入力ENFが開けない
12	入力ENFが不正
15	エラー検出結果ファイル(.csv)を開けない
18	出力ENFが開けない
21	出力ENFが書き込めない
24	XMLログが書き込めない
<0	負の値 ライセンスエラー

II. 実行結果

PDQ Checker の実行結果は、以下のレポートで確認することができます。

- チェック結果CSVファイル

PDQ checkerを実行すると、以下のCSVファイルが実行結果として出力されます。

```

[ヘッダー]
バージョン情報  ...PDQ checkerのバージョン情報
ファイル名     ...対象ファイル名
実行時刻      ...実行した時刻

[検証結果]
-- ファイル名: 「対象ファイル名」 --  ...モデル全体の結果を出力します。

[項目ID] [項目名] [しきい値] [エラー数] [重要度]
-- ボディ名: 「ボディ名」 --  ...各ボディの結果を出力します。

[項目ID] [項目名] [しきい値] [エラー数] [重要度] [Min] [Max]

[Min][Max]  ...エラーとなる要素について、そのエラー値の最小値と最大値

```

- チェック結果PDF レポート

以下のパラメータを設定した場合、チェック結果が 3D PDF 形式のレポートとして出力されます。

```
Create3DPdfReport=1
```

補足

- 3D PDF レポートを出力するためには、enf23dpdf ライセンスが必要となります。
- 3D PDF レポートのファイル名は CSV レポートの拡張子を .pdf に変えたものとなります。
- 3D PDF レポートの出力は XML シナリオでの実行にのみ対応しています。引数からの実行および checknut.exe の単独実行には対応していません。
- [ENFCHECKER_PriorityMapping] に "重度"("Critical")、"中度"("Serious")、"軽度"("Minor") 以外を指定した場合、すべて "中度"("Serious") として扱われます。
- パラメータ "Create3DPdfReport_Type=1" を指定すると、カスタマイズした形式のレポートを出力することができます。この場合、enf23dpdf ライセンスの他に pdfeditor ライセンスが必要となります。カスタマイズの詳細については弊社までお問い合わせください。

III. 項目名、重要度の定義

検証結果の CSV ファイルに出力される、項目名称、重要度はデフォルトのものが設定されますが、パラメータファイルによりカスタマイズする事も可能です。

- (記述例)

```
[ENFCHECKER_NameMapping]
G-L0-IS = エッジループの自己干渉(必須項目)
G-FA-IS = エッジループ間の干渉(任意)
[ENFCHECKER_PriorityMapping]
G-L0-IS = 修正必須
G-FA-IS = 修正任意
```

IV. PDQ チェック項目の設定

付属の「PDQCheckSetting.xls」を開きます(**\util\PdqChkr)。変換元のCADと変換先のCADを指定した後、チェックするPDQ項目に"1"を指定し、作成ボタンを押す事で、パラメータファイルを作成します。

また、薄板検出の機能が用意されています。パラメータファイルに下記を追加することで、薄板を検出することが可能です。各パートについて、ある任意の方向についての厚みがしきい値以下であるとき薄板として検出し、その位置情報をXMLログに記述します。
検出するしきい値は、Tol_ThinPartで指定します。単位はmmです。

```
[ENFCHECKER]
CheckThinPart=1
Tol_ThinPart=1
```

V. 注意点 (引数"-n"で実行する場合)

以下の条件に応じて、結果ファイル(.csv)がワークディレクトリに出力されます。

1.	停止オプション、ヒーリングスキップオプションの指定無し
	[入力ファイル名].csv: ヒーリング前のチェック結果 [出力ファイル名].csv: ヒーリング後のチェック結果
2.	停止オプション "-s -b" を指定した場合
	[入力ファイル名].csv: ヒーリング前のチェック結果
3.	停止オプション "-s -a" で、入力ファイルが CADファイルの場合
	[入力ファイル名].csv: ヒーリング前のチェック結果 [出力ファイル名].enf.csv: ヒーリング後のチェック結果
4.	停止オプション "-s -a" で、入力ファイルが ENFファイルの場合
	[出力ファイル名].enf.csv: ヒーリング後のチェック結果
5.	ヒーリングスキップオプション "-k" で、入力ファイルに ENFファイルを指定した場合
	結果ファイルは出力されません。

VI. XMLログ内容

出力結果XMLには、下記の情報が出力されます。

- PDQオプション (検証時の設定内容 : 検証対象の項目やしきい値などの情報)
- PDQエラー (PDQ検証結果)

- XMLログの書式 (PDQオプション)

ASFALISXMLログ仕様 (ASFALISLOG.pdf) をご参照ください。

基本的な構成は、ASFALIS XMLログ仕様 4章 に記載されたものに従い、追加で5章に記載された要素を出力します。

VII. しきい値の設定についての注意点

PDQCheckerの内部では、下記の検証項目に関して、内部的に別の検証項目の検証しきい値を利用して検証していますので、ご注意ください。

- セグメント間の折れ
セグメント間の折れの検証対象を、セグメント間に隙間がない場合に限定しており、その選定に「セグメント間の隙間」のしきい値を利用している。
- セグメント間の曲率不連続
セグメント間の曲率連続性の検証対象を、セグメント間に隙間がなく、かつ折れがない場合に限定しており、その選定に「セグメント間の隙間」「セグメント間の折れ」のしきい値を利用している。
- サーフェスパッチ間の折れ
サーフェスパッチ間の折れの検証対象を、サーフェスパッチ間に隙間がない場合に限定しており、その選定に「サーフェスパッチ間の隙間」のしきい値を利用している。
- サーフェスパッチ間の曲率不連続
サーフェスパッチ間の曲率連続性の検証対象を、サーフェスパッチ間に隙間がなく、かつ、折れがない場合に限定しており、その選定に「サーフェスパッチ間の隙間」「サーフェスパッチ間の折れ」のしきい値を利用している。
- フェース間の折れ
フェース間の折れの検証対象を、フェース間に隙間がない場合に限定しており、その選定に「フェース間の隙間」のしきい値を利用している。
- フェース間の曲率不連続
フェース間の曲率連続性の検証対象を、フェース間に隙間がなく、かつ折れがない場合に限定しており、その選定に「フェース間の隙間」「フェース間の折れ」のしきい値を利用している。

3.3.2. Geometry Simplifier

指定されたファイルを自動で簡略化するコンポーネントです。簡略化の機能として下記が実行可能です。

微小エッジの削除機能	モデル中に含まれる微小なエッジを消去した上で、離れた部分をつなぐなど位相構造を更新します。
微小フェースの削除機能	モデル中に含まれる微小なフェースを消去した上で、離れた部分をつなぐなど位相構造を更新します。

フェースのマージ機能	モデル中に含まれる複数枚のフェースを接続して一枚のフェースにします。
丸穴の除去機能	モデル中に含まれる丸穴を認識して消去します。
一般穴の除去機能	モデル中に含まれる一般穴 (丸穴以外) を認識して消去します。
シート穴の除去機能	厚みのないシートモデル中に含まれる穴を認識して消去します。
フィレットの除去機能	モデル中に含まれるフィレットを認識して消去します。
面取り除去機能	モデル中に含まれる面取りを認識して消去します。
段差の除去機能	モデル中に含まれる段差を認識して消去します。
丸穴の抽出機能	モデル中に含まれる丸穴の空洞形状を抽出します。
一般穴の抽出機能	モデル中に含まれる穴 (丸穴以外) の空洞形状を抽出します。
ボス・リブの除去機能	モデル中に含まれるボス/リブを認識して消去します。
リブの除去機能	モデル中に含まれるリブを認識して消去します。
溝の除去機能	モデル中に含まれる溝を認識して消去します。
ロゴの除去機能	モデル中に含まれるロゴを認識して消去します。
曲線群の平滑化機能	エッジ同士が折れて接続しているときに、滑らかにつながるようにエッジを再計算します。
微小パートの除去機能	モデル中に含まれる微小なパートを認識して消去します。
微小ボリウムの除去機能	モデル中に含まれる微小なボリウムを認識して消去します。
見えないボリウムの除去機能	モデル中に含まれる外部から見えないボリウムを認識して消去します。
穴溝突起物の一括削除	モデル全体をおおよそ覆う面を基準として、その面に覆われる穴・溝と、その面から一定範囲内に収まる突起物を一括で認識して消去します。
簡易形状置換	ボリウム、パート、またはアセンブリを簡易形状 (直方体、円柱、円筒、押し出し形状) に置換します。
ソリッド外形抽出機能	全てのパートが干渉または接触しているアセンブリモデルを1ソリッドとして抽出します (ソリッド間に隙間があるモデルは1ソリッドになりません)。
可視面抽出機能	指定した視点方向から見えるフェースのみを抽出します。

I. Geometry Simplifierの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

II. 簡略化項目の設定

パラメータファイルに簡略化手順と項目を記述します。

* 詳細は、ASFALIS Parameterマニュアルを参照してください。

- (例)


```

Process_0=RemoveFillet // 1番目 フィレットの削除
FilletMaxRadius_0=2.5 // しきい値 半径 2.5mm 以下のフィレット
Process_1=ExportENF
Process_2=RemoveRoundHole // 2番目 丸穴の削除
RoundHoleMaxDiameter_2=10.0
Process_3=ExportENF
Process_4=MergeFace // 3番目 フェースのマージ
MergeAngleTolerance_4=5
ProcessNumberMax=5 // 実行手順数

```

III. 干渉回避機能

フィレット除去の結果、アセンブリのパート間で干渉が発生する場合に処理をもとに戻す機能です。内部的に干渉チェックのモジュールを呼び出して検証を行いつつフィレット除去を実行します。パラメータの設定方法につきましては、パラメータマニュアルをご参照ください。

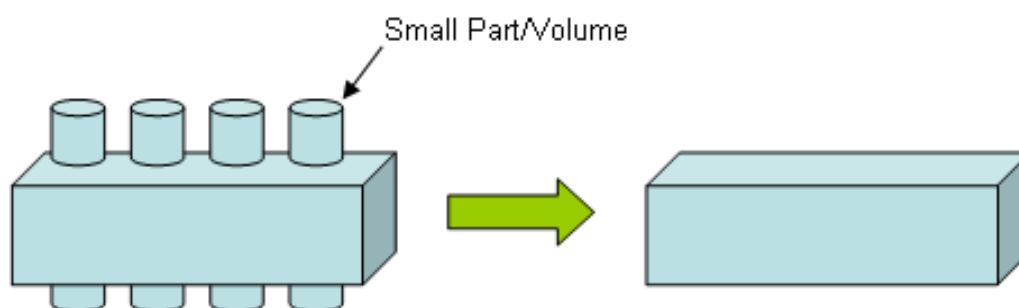
干渉回避機能使用時は、コンフィグファイルにキーワードINTRCHECKを設定してください。

- (例)

```
INTRCHECK    C:\elysium\bin\Tools\win\elyIntrCheck\IntrCheck
```

IV. 微小パート/ボリウムの除去機能

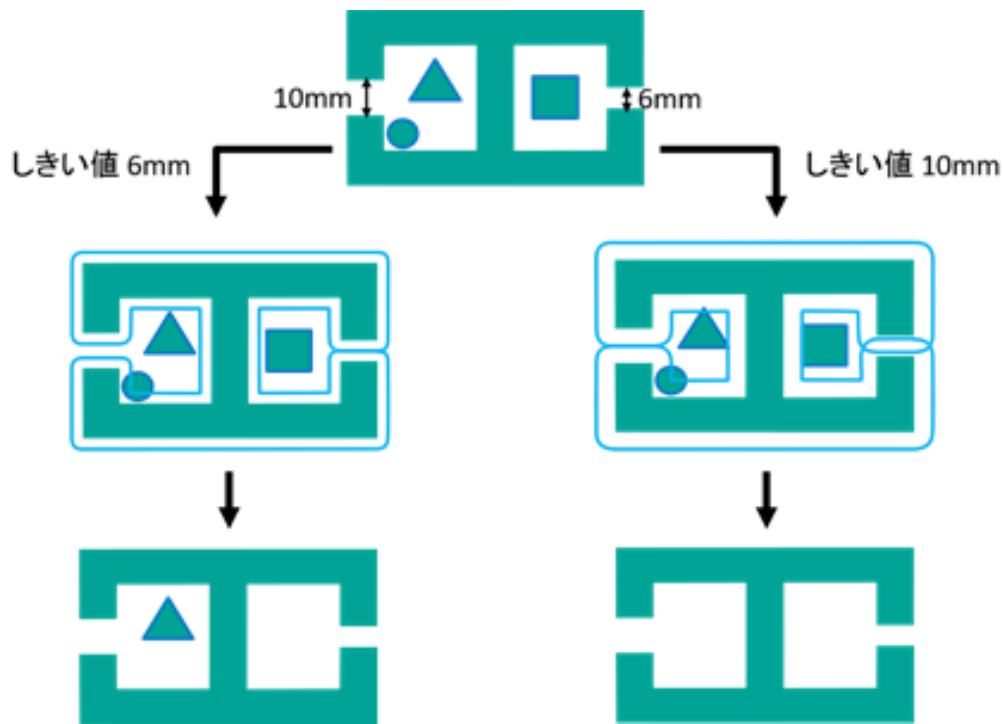
モデル中に微小とみなせるパート/ボリウムが存在する場合、微小パート/ボリウムを削除します。パート内の全ボリウムが削除対象となった場合、パートが削除されますのでアセンブリ構造が変わります。微小とみなすパート/ボリウムの条件はパラメータで設定します。パラメータの設定方法につきましては、パラメータマニュアルをご参照ください。



V. 見えないパートの除去機能

外部から見えないボリウムを一括して削除します。その際、指定された値以下の隙間は無視して(埋まっているものとして)判定を行います。パート内の全ボリウムが削除対象となった場合、パートが削除されるためアセンブリ構造が変わります。

見えないパートの削除機能(RemoveInvisiblePart)および可視面抽出機能(ExtractVisibleFace)を使用する場合、実行環境としてバージョン3.0以上のOpenGLに対応したビデオボードのある環境を推奨します。GPUがない環境でも動作しますが、その場合、EX7.2までの動作と同等となります。



3.3.3. CAD Validator

2つのENFファイルと比較し、比較結果をModel Viewerで高速表示可能なdvfx形式、もしくはCADdoctor、Model Viewerで読み込み可能なenf_diff形式のファイルで保存します。また比較結果を各種レポートファイルとして出力することもできます。レポートの詳細については [VI. 各種レポートファイル](#) をご参照ください。

I. CAD Validatorの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

I.I. 2つのCADファイルを比較実行するシナリオ

ASFALIS SmartLauncher、ASFALISコントローラ、ASFALIS TransServer向けに、代表的なCADについて2つのCADファイルを入力にして比較実行するシナリオを用意しました。

<Baseパッケージ>\sample\geomdiff

- adc_scenario: ASFALIS SmartLauncher、ASFALISコントローラ向けシナリオ
- ats_scenario: ASFALIS TransServer向けシナリオ

I.I.I. ASFALIS SmartLauncher 向けシナリオ

ASFALIS SmartLauncher を導入した環境にて、該当するシナリオファイルをシナリオフォルダに置いて使用してください(.rb, .adc の2種類のファイルで1セットとなっています)。シナリオフォルダはASFALIS SmartLauncher のシステム設定から確認、変更することができます。

ASFALIS SmartLauncherのWindows コンテキストメニューで「シナリオ実行」を選択した際に表示される「シナリオ実行」ウィンドウでシナリオと入力モデルを選択して実行してください。モデルは2つ指定する必要があります。

3D PDF形式の比較レポートファイルを出力するためには、シナリオファイルに "Create3DPdfReport=1" を設定する必要があります。事前にシナリオファイル(.rb ファイルのみ) に以下の通り設定してください(既定値の "0" を "1" に変更してください)。

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

I.I.II. ASFALISコントローラ向けシナリオ

ASFALISコントローラを導入した環境にて、該当するシナリオファイルを以下のフォルダに置いてご使用ください(.rb, .adc の2種類のファイルで1セットとなっています)

<ASFALISコントローラ インストールフォルダ>\data\scenario

※ASFALISコントローラのメインダイアログの「シナリオ」から選択できるようになります。

※「入力」には2つCADファイルを指定して実行してください。

3D PDF形式の比較レポートファイルを出力するためには、シナリオファイルに "Create3DPdfReport=1" を設定する必要があります。事前にシナリオファイル(.rb ファイルのみ) に以下の通り設定してください(既定値の "0" を "1" に変更してください)。

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

なお、このシナリオはマニュアル編集したものであるため、ASFALISコントローラのメインダイアログ上で編集することはできません。

I.I.III. ASFALIS TransServer向けシナリオ

該当シナリオファイルをASFALIS TransServer にインポートしてご使用ください。

シナリオ一覧画面の「インポート」ボタンからシナリオファイル(.json ファイル) をアップロードすることでインポートできます。

シナリオは "validate from <Source側CAD名> to <Target側CAD名>" という名前で作成されます。これをジョブ実行時に選択してご使用ください。

3D PDF形式の比較レポートファイルを出力するためには、Create3DPdfReport=1 のパラメータを設定する必要があります。ジョブ実行時に比較用のパラメータ設定にて「比較レポートファイル (3D PDF形式) 作成」をONにしてください。

II. 比較項目の設定

パラメータファイルに比較する項目を設定します。詳細は、ASFALIS Parameterマニュアルを参照してください。

(グループフラグ定義ファイルについて)

CAD Validator の出力する 3D PDF レポートで差異をグループ化した際、グループ単位で差異についての判定フラグを設定することができます。既定で設定できるフラグは OK と Reject です。グループフラグ定義ファイルを使用することで、任意のフラグを設定することができるようになります。このファイルは以下の規則に従って記述してください。

- JSON 形式で記述してください。
- フラグオブジェクトの配列を定義してください。
 - 配列中のオブジェクト一つごとにフラグが一つ定義されます。
 - 配列の長さは 100 以下にしてください。
- フラグオブジェクトのプロパティには以下の内容を定義してください。
 - choices (mandatory)
 - フラグの選択肢としてコンボボックスに表示される選択肢を指定します。
 - 型は String の Array です。長さは任意です。(例: ["OK", "NG", "Postpone"])
 - name (mandatory)
 - フラグの選択インターフェース上で表示されるフラグ名を指定します。
 - 型は String です。(例: "承認フラグ")
 - approvedChoices (optional)
 - choices と連結されてコンボボックス中の選択肢として表示されます。このプロパティで指定された選択肢が選択されており、かつ「差異要素のみ表示」チェックボックスにチェックが入っている場合、グループテーブルに表示されなくなります。
 - フラグオブジェクトの配列のうち、一つ以下のフラグオブジェクトがこのプロパティを持つようにしてください。
 - 型は String の Array です。長さは任意です。(例: ["優", "良", "可"])

approvedChoices に ["優", "良", "可"] が定義され、choices に ["不可", "保留"] が定義されているとき、コンボボックスには両者を併せた5つの選択肢が表示されます。approvedChoices で指定された三種類の選択肢のいずれかが選択された場合、そのグループは承認されたものとみなされます。この場合 Diff や Reject の要素がグループに属していたとしても「差異のみ表示」が有効な場合には、Same や OK の要素の様に表示されなくなります。

以下は、グループフラグ定義ファイルのサンプルです。

サンプル1(既定の設定)

```
[{name: "承認フラグ", choices: ["Reject"], approvedChoices: ["OK"]}]
```

サンプル2

```
[
  {
    "name": "品質",
    "choices": [
      "A",
      "B",
      "C"
    ]
  },
  {
    "name": "確認結果",
    "choices": [
      "判定不能",
      "不可"
    ],
    "approvedChoices": [
      "可",
      "良",
      "優",
    ]
  },
  {
    "name": "対応予定者",
    "choices": [
      "UserA",
      "UserB",
    ]
  },
],
]
```

III. XMLログについて

CAD ValidatorでXMLログを出力した時の形式について以下に説明します。設定した場合のみログに出力されます。

<CompareResultSummary>

- 比較結果の合計

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. <sourcefile> (比較元モデル)で指定したENFに含まれるパートの数
2. <targetfile> (比較先モデル)で指定したENFに含まれるパートの数
3. 詳細比較および高速比較で比較できないため削除した比較元ポリゴンパート数
4. 詳細比較および高速比較で比較できないため削除した比較先ポリゴンパート数
5. 比較されたパートの数(比較元、比較先を別カウントするので必ず偶数)
6. 詳細比較のエッジ群間の位置比較の結果、最小検出距離以上の離れのあるエッジを1つ以上含む比較元のパート数 (オプション ComparisonMode = 0かつCheckEdge = 1で比較を実行した場合のみ有効)
ComparisonMode = 1の場合はフリーエッジのみを比較対象とする。
7. 詳細比較のフェース群間の位置比較の結果、最小検出距離以上の離れのあるフェースを1つ以上含む比較元モデルのパート数 (オプション ComparisonMode = 0かつCheckFace = 1で比較を実行した場合のみ有効)
8. 連続性の比較の結果、連続性の変化があるエッジを1つ以上含む比較元モデルのパート数 (オプション ComparisonMode = 0/CheckCont = 1で比較を実行した場合のみ有効)
9. 位相比較の結果、全く対応する要素が見つからないエッジを1つ以上含む比較元モデルのパート数 (オプション ComparisonMode = 0/CheckTopology = 1で比較を実行した場合のみ有効)
10. 位相比較の結果、全く対応する要素が見つからないフェースを1つ以上含む比較元モデルのパート数 (オプション ComparisonMode = 0/CheckTopology = 1で比較を実行した場合のみ有効)
11. 位相比較の結果、全く対応する要素が見つからないエッジを1つ以上含む比較先モデルのパート数 (オプション ComparisonMode = 0/CheckTopology = 1で比較を実行した場合のみ有効)
12. 位相比較の結果、全く対応する要素が見つからないフェースを1つ以上含む比較先モデルのパート数 (オプション ComparisonMode = 0/CheckTopology = 1で比較を実行した場合のみ有効)
13. 高速比較の結果、UniqueFaceが検出されたパートの数 (オプション ComparisonMode=1で比較を実行した場合のみ有効)
14. 比較に失敗した要素を1つ以上含むパート数
15. 構成比較の結果で、比較元、比較先間でペアを作れなかった要素の数 (比較元と比較先の和)
16. 構成比較の結果で、名前に差異がある要素ペアの数
17. 構成比較の結果で、PartName 属性に差異がある要素ペアの数
18. 構成比較の結果で、PartNumber 属性に差異がある要素ペアの数
19. 構成比較の結果で、アセンブリモデル中の配置に差異がある要素ペアの数
20. 構成比較の結果で、親アセンブリが異なる要素ペアの数
21. 構成比較の結果で、形状差異がある要素ペアの数
22. 高速表示用比較で、比較に失敗したエッジの数
23. 高速表示用比較で、比較に失敗したフェースの数
24. 高速表示用比較で、フェースの形状差異が検出された数
25. 高速表示用比較で、単独カーブの形状差異が検出された数
オプション CheckEdge = 1で比較を実行した場合のみ有効

26. 高速表示用比較で、フリーエッジの形状差異が検出された数
オプション CheckEdge = 1 で比較を実行した場合のみ有効
27. 高速表示用比較で、単独点の形状差異が検出された数
オプション CheckVertex = 1 で比較を実行した場合のみ有効
28. 高速表示用比較で、フェース法線の形状差異が検出された数
オプション CheckAngleFaceNormal = 1 で比較を実行した場合のみ有効
29. 高速表示用比較で、ポリゴンの形状差異が検出された数
オプション CheckPolygon = 1 で比較を実行した場合のみ有効
30. 高速表示用比較で、ポリラインの形状差異が検出された数
オプション CheckPolyline = 1 で比較を実行した場合のみ有効
31. 高速表示用比較で、ポリゴンポイントの形状差異が検出された数
オプション CheckPolygonPoint = 1 で比較を実行した場合のみ有効
32. 高速表示用比較で、システム属性の差異が検出された数
オプション CheckSystemProperty = 1 で比較を実行した場合のみ有効
33. 高速表示用比較で、ユーザ属性の差異が検出された数
オプション CheckUserProperty = 1 で比較を実行した場合のみ有効
34. 高速表示用比較で、フェースの属性の差異が検出された数
オプション CheckAttributeFace = 1 で比較を実行した場合のみ有効
35. 高速表示用比較で、単独カーブの属性の差異が検出された数
オプション CheckAttributeIsolatedEdge = 1 で比較を実行した場合のみ有効
36. 高速表示用比較で、単独点の属性の差異が検出された数
オプション CheckAttributeIsolatedVertex = 1 で比較を実行した場合のみ有効
37. 高速表示用比較で、注記の差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
38. 高速表示用比較で、データムの差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
39. 高速表示用比較で、データムターゲットの差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
40. 高速表示用比較で、サーフェス仕上げの差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
41. 高速表示用比較で、幾何公差の差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
42. 高速表示用比較で、寸法の差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
43. 高速表示用比較で、ロケータの差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
44. 高速表示用比較で、スポット溶接の差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効

45. 高速表示用比較で、ライン溶接の差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効
46. 高速表示用比較で、モデルビューの差異が検出された数
オプション CheckPMI = 1 で比較を実行した場合のみ有効

<GeomDiff>

- 要素の距離比較

```

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" />

```

- 計測距離上限以上の離れがあったエッジのパートIDとエッジIDと離れの検出位置(曲線のパラメータ)
該当エッジのパート内での座標(代表値1点)
オプション ComparisonMode = 0/CheckEdge = 1 で比較を実行した場合に有効。
ComparisonMode=1の場合はフリーエッジを対象として検出される。
- 検出範囲の離れがあったエッジのパートIDとエッジIDと離れの検出位置(曲線のパラメータ)
2つめのDekEntityは、1つめのDekEntityに対する最近点の情報(この2点間の距離がdist)
該当エッジのパート内での座標(代表値1点)

オプション ComparisonMode = 0/CheckEdge = 1 で比較を実行した場合に有効。

ComparisonMode=1の場合はフリーエッジを対象として検出される。

3. 左側モデル中で位置差異が検出された全エッジの数(outOfRangeと検出範囲のものの合計)
オプション ComparisonMode = 0/CheckEdge = 1 で比較を実行した場合のみ有効。
4. 計測距離上限以上の離れがあったフェースのパートIDとフェースIDと離れの検出位置(曲面のパラメータ)
該当フェースのパート内での座標(代表値1点)
オプション ComparisonMode = 0/CheckFace = 1 で比較を実行した場合のみ有効。
5. 検出範囲の離れがあったフェースのパートIDとフェースIDと離れの検出位置(曲面のパラメータ)
該当フェースのパート内での座標(代表値1点ずつ)
2つめのDekEntityは、1つめのDekEntityに対する最近点の情報(この2点間の距離がdist)
オプション ComparisonMode = 0/CheckFace = 1 で比較を実行した場合のみ有効。
6. 左側モデル中で位置差異が検出された全フェースの数(outOfRangeと検出範囲のものの合計)
オプション ComparisonMode = 0/CheckFace = 1 で比較を実行した場合のみ有効。

<BodyIdToComponentIdMap>

- bodyのIdとcomponentのIdの対応

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

- Id=1のBodyとId=1のComponentが対応していることを示す

<ElementCorre>

- 位相比較の結果(要素の対応関係)、高速比較の結果、およびCAD比較でのエラー情報

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." />
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. ID=2のパートのID=6のフェースが比較失敗したことを示す。
失敗したフェースを特定するための頂点のローカル座標。
比較失敗となった原因説明。
2. ID=1のパートが比較元モデルのパートであることを示す(bodyDekIdとdekIdは常に同じ)
3. ID=2のパートに比較対象のパートが存在することを示す(bodyDekIdとdekIdは常に同じ)
elementCorreIDの値が等しいElementCorreGrpがもう1つ存在し、そのパートと比較が行われている
4. パートID=2でエッジID=6のエッジには、1対1で対応するエッジが存在することを示す
elementCorreIDの値が等しいElementCorreGrpがもう1つ存在し、そのエッジと対応がある
(オプション ComparisonMode = 0/CheckTopology= 1 で比較を実行した場合のみ有効)
5. パートID=2でフェースID=17のフェースには、1対1で対応するフェースが存在することを示す
elementCorreIDの値が等しいElementCorreGrpがもう1つ存在し、そのフェースと対応がある
(オプション ComparisonMode = 0/CheckTopology= 1 で比較を実行した場合のみ有効)
6. パートID=2でフェースID=12と10の2フェースには、M対Nで対応するフェースが存在することを示す
elementCorreIDの値が等しいElementCorreGrpがもう1つ存在し、そのフェース群と対応がある
オプション ComparisonMode = 0/CheckTopology = 1 で比較を実行した場合のみ有効
(エッジの例はないがcorrespondenceEdgeMtoNはフェースと同様の仕様)
7. パートID=1でエッジID=4のエッジには、1対1でもM対Nでも対応する要素がないことを示す

(オプション ComparisonMode = 0/CheckTopology = 1 で比較を実行した場合のみ有効)

8. パートID=2でフェースID=1のフェースには、1対1でもM対Nでも対応する要素がないことを示す
(オプション ComparisonMode = 0/CheckTopology = 1 で比較を実行した場合のみ有効)
9. 1対1でもM対Nでも対応する要素がないエッジの総数(比較元+比較先モデル)
(オプション ComparisonMode = 0/CheckTopology = 1 で比較を実行した場合のみ有効)
10. 1対1でもM対Nでも対応する要素がないフェースの総数(比較元+比較先モデル)
(オプション ComparisonMode = 0/CheckTopology = 1 で比較を実行した場合のみ有効)
11. ComparisonMode=1 の時に検出された対応する要素がないフェース群のパート毎の数
(オプション ComparisonMode = 1 で比較を実行した場合のみ有効)
12. ComparisonMode=1 の時に検出された対応する要素がないフェース群の総数(比較元+比較先モデル)
(オプション ComparisonMode = 1 で比較を実行した場合のみ有効)

<ContDiff>

- 連続性

```
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. contDiffType=2は、比較元モデルで検出折れ角度以下の折れだった箇所に対して、比較先モデルで折れが発生したことを示す
1つ目のContDiffElemは比較元モデルのエッジID=8、2つ目のContDiffElemは比較先モデルのエッジID=8の位置を示す
パラメータと折れ角度は比較元モデル(折れがあるほう)のエッジについての記述
(オプション ComparisonMode = 0/CheckCont = 1 で比較を実行した場合のみ有効)
2. contDiffType=3は、比較先モデルで検出折れ角度以上の折れだった箇所が比較元モデルでは折れが解消したことを示す
1つ目のContDiffElemは左側モデルのエッジID=25、2つ目のContDiffElemは比較先モデルのエッジID=25の位置を示す
パラメータと折れ角度は比較先モデル(折れがあるほう)のエッジについての記述
(オプション ComparisonMode = 0/CheckCont = 1 で比較を実行した場合のみ有効)
3. 全モデル内で連続性の変化(折れの発生+解消)があった位置(エッジ単位)の数
(オプション ComparisonMode = 0/CheckCont = 1 で比較を実行した場合のみ有効)

IV. 制限事項

CAD Validator はUNCパスにインストールして実行することはできません。

V. カスタマイズファイルの設定

高速表示用比較の際、比較対象要素の追加/削除、パート/アセンブリのマッピングロジック、比較ロジック等を細かく制御することができます。それらはカスタマイズファイルによって指定します。

このカスタマイズファイルでは、PMI、属性、B-rep属性の比較について制御することができます。形状の比較についてはパラメータファイルで制御します。

カスタマイズファイルは、customize_table.csv、customize_utility.rbの2ファイルで構成されており、以下のパラメータで指定します。

CustomizeFolder=<フォルダパス>

- customize_table.csv
テーブル形式の設定ファイルで、以下の内容を指定することができます。
 - 比較対象要素、プロパティの追加/削除
 - パート/アセンブリ、要素のマッピングロジックの変更
 - プロパティ毎の比較ロジックの変更
 - プロパティ毎の比較トレランスの変更
- customize_utility.rb
マッピングロジック、比較ロジックが複雑な場合、詳細な処理をこのファイルに記述し、customize_table.csvではその処理を参照することができます。
テーブル形式の customize_table.csv のみで記述するより高いメンテナンス性を維持することができます。

customize_table.csv は詳細に比較設定を制御できますが、比較対象要素の追加/削除についてはパラメータでも切り替えられるようにしております。

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

また、customize_table.csv と customize_utility.rb は比較対象のCADごとに推奨の設定が異なりますが、代表的なCADについては推奨の初期設定を用意しております。

(参照) <ASFALIS Components>\tools\win\cadvalidator\template\

推奨の初期設定ではうまく比較できない場合、カスタマイズファイルを独自に用意する必要があります。カスタマイズファイルを詳細に編集する場合は購入元までお問い合わせください。

VI. 各種レポートファイル

CAD Validator では、以下の表にあるレポートファイルを出力することができます。いずれも閲覧時にライセンスは必要ありません。

レポートの種類	2D XML形式	3D HTML形式	3D PDF形式	ビューア形式
形状差異の表示	○	○	○	○
PMI差異の表示	×	○	○	×
属性差異の表示	×	○	○	×
大規模データを表示した際の応答性	×	○	△	○
対話的ビューイング	×	○	○	○
コメントの追加	×	×	○	○
ファイル形式	複数ファイル	複数ファイル	単一ファイル	単一ファイル
環境/前提条件	Internet Explorer 11	Internet Explorer 11 (*1)	Acrobat Reader DC以降	Windows 8.1以降の64bit OS (*2)

- (*1) ActiveX コントロールを有効にする必要があります。
- (*2) ビューア形式レポートのexeファイルにはデジタル署名がされていません。OSのセキュリティ設定によっては閲覧するために管理者権限が必要になる場合があります。

それぞれのレポートの詳細については次ページ以降を参照してください。

CAD比較レポートファイル(XML形式)

シナリオXMLに定義することで、形状比較結果のレポートファイルを出力することが可能です。

出力方法

パラメータファイルでCreateReport=1を指定します。

XMLシナリオの<xmlreportfile>タグで出力するレポートファイル名を直接指定することができます。<xmlreportfile>タグを指定しない場合は、以下の優先順でファイル名を自動決定します。

1. <diffresultfile>タグがある場合：
<diffresultfile>タグのファイル名部分 + "_report.xml"
2. <diffresultfile>タグがない場合：
<xmllogfile>タグのファイル名部分 + "_report.xml"

出力ファイル構成

<xmlreportfile>タグで指定した.xmlファイルの他に、補助的なファイルが同一フォルダに作成されます。

レポートファイル名がReportFileName.xmlである場合、出力ファイルは以下のとおりです。

ReportFileName.xml	レポート本体のファイル
ReportFileName	キャプチャされた画像、視点ファイル等を格納するフォルダ (レポート本体と同じフォルダ名)
elyReportComp***.xsl	xmlをブラウザで表示可能なフォーマットに変換するためのファイル (***)部分はレポートのバージョン・種類ごとに異なります)

出力したレポートを別の場所に移動するには、ReportFileName.xmlを移動するだけでなく、ReportFileNameフォルダとelyReportComp***.xslファイルも一緒に移動する必要があります。

• 視点ファイル

詳細比較および高速比較実行時は、レポートファイルと同名のフォルダに視点ファイルが作成されます。作成された視点ファイルをModel ViewerおよびCADdoctorにドラッグ&ドロップすることで、レポートの画像と同一視点を再現することができます。

高速表示用比較の場合は、視点ファイルは作成されません。

• 動作・制限等

- 処理対象のモデルによっては、比較処理よりもレポート出力処理の方が CPU およびメモリを消費する場合があります。
- 比較処理終了後にレポート作成が実行されますので、パフォーマンスの問題等でレポート出力が失敗する場合でも、<diffresultfile>及び<xmllogfile>の指定による比較結果は得られます。
- 出力ファイルと同じ.xmlファイル、もしくは画像フォルダが既に存在する場合は、レポート書き出しをスキップします。
- 出力されたレポートファイルは、Internet Explorer 11 で表示することが可能です。それ以外のブラウザは動作保証外です。
- <xmlreportfile>と<html3dreportfile>に拡張子以外が同じ名前のファイルパスを指定することはできません。CAD比較レポートファイル(XML形式)と形状比較レポートファイル(3D HTML形式)はいずれもサブフォルダを作りますが、そのフォルダ名が重なるためです。
- <xmlreportfile>で指定したファイルが既に存在する、または拡張子を除いたファイル/フォルダが既に存在する場合、レポートファイルは保存されません。

比較レポートファイル (3D HTML形式)

• 出力方法

パラメータファイルでCreate3DReport=1 を指定します。

XMLシナリオの<html3dreportfile>タグで出力するレポートファイル名を直接指定することができます。<html3dreportfile>タグを指定しない場合は、以下の優先順でファイル名を自動決定します。

1. <diffresultfile>タグがある場合：
<diffresultfile>タグのファイル名部分 + "_3dreport.html"
2. <diffresultfile>タグがない場合：
<xmllogfile>タグのファイル名部分 + "_3dreport.html"

• 出力ファイル構成

<html3dreportfile>タグで指定した.htmlファイルの他に、補助的なファイルが同一フォルダに作成されます。

レポートファイル名がReportFileName.htmlである場合、出力ファイルは以下のとおりです。

ReportFileName.html	レポートのトップファイル
ReportFileName	レポート本体のファイル、JavaScript、CSS、比較結果のデータファイル等を格納するフォルダ(レポート本体と同じフォルダ名)

出力したレポートを別の場所に移動するには、ReportFileName.htmlを移動するだけでなく、ReportFileNameフォルダも一緒に移動する必要があります。

• 動作・制限等

- 出力されたレポートファイルは、Internet Explorer 11 で表示することが可能です。それ以外のブラウザは動作保証外です。
- インターネットに接続していない環境でレポートを表示したい場合は、以下のパラメータを立てて比較を実行してください。
Enclose3DReportAddon=1
比較レポートファイル(3D HTML形式)のファイル構成に RegisterFlgControlGV.bat が追加されるので、レポートを開覧する環境にてこのバッチファイルを管理者権限で実行してください。
- <xmlreportfile> と <html3dreportfile> に拡張子以外が同じ名前のファイルパスを指定することはできません。CAD比較レポートファイル(XML形式)と形状比較レポートファイル(3D HTML形式)はいずれもサブフォルダを作りますが、そのフォルダ名が重なるためです。
- <html3dreportfile>で指定したファイルが既に存在する、または拡張子を除いたファイル/フォルダが既に存在する場合、レポートファイルは保存されません。

比較レポートファイル (3D PDF形式)

• 出力方法

パラメータファイルでCreate3DPdfReport=1を指定します。

XMLシナリオの<pdf3dreportfile>タグで出力するレポートファイル名を直接指定することができます。<pdf3dreportfile>タグを指定しない場合は、以下の優先順でファイル名を自動決定します。

- <diffresultfile>タグがある場合：
<diffresultfile>タグのファイル名部分 + "_3dpdfreport.pdf"
- <diffresultfile>タグがない場合：
<xmllogfile>タグのファイル名部分 + "_3dpdfreport.pdf"

• 動作・制限等

- <pdf3dreportfile> で指定したファイルが既に存在する場合、レポートファイルは保存されません。
- レポートの出力は比較処理と比べて高負荷な機能です。
- 3D PDF形式で出力する際に PDF Editor を併せて利用する場合には [3.3.9, “3D PDF Editor”](#)

もご参照ください。

ビューア形式レポートファイル

- 出力方法

パラメータファイルでCreateDiffAnalyzer=1を指定します。

XMLシナリオの<diffanalyzer>タグで出力するレポートファイル名を直接指定できます。

<diffanalyzer>タグを指定しない場合は、以下の優先順でファイル名を自動決定します。

1. <diffresultfile>タグがある場合：
<diffresultfile>タグのファイル名部分 + "_diffanalyzer.exe"
2. <diffresultfile>タグがない場合：
<xmllogfile>タグのファイル名部分 + "_diffanalyzer.exe"

- 動作・制限等

- Diff-Viewer と同等の機能がある自己実行形式の exe ファイルです。特定の比較結果を表示します。
- 本レポートを閲覧する上で特定のアプリケーションをインストールする必要はありません。また追加のライセンス等も必要ありません。
- 比較処理終了後にレポート作成が実行されますので、パフォーマンスの問題等でレポート出力が失敗する場合でも、<diffresultfile>及び<xmllogfile>の指定による比較結果は得られます。
- <diffanalyzer> で指定したファイルが既に存在する場合、レポートファイルは保存されません。
- 本レポートを実行するための推奨環境は Model Viewer の推奨環境と同様です。Model Viewer の推奨環境については、以下のパスにあるリリースノートをご参照ください。
<ASFALIS Base パッケージ>\doc\ASFALIS_Model_Viewer

- 使用方法

本レポートの使用方法についてはヘルプを参照してください。ヘルプはレポートを起動した際に画面右上に表示される「？」アイコンをクリックすると表示されます。

3.3.4. Attribute Editor

ENFの属性情報の操作を行なうコンポーネントです。以下の操作を実行する事が可能です。

- [属性のマッピング](#)
- [属性の抽出](#)
- [属性値の置換](#)
- [属性値の追加](#)
- [インスタンス名の編集](#)

I. Attribute Editorの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

- シナリオ例

```
<?xml version="1.0" encoding="Shift_JIS"?>
<ScenarioList>
  <Scenario>
    <!-- CADからENFに変換する -->
    <CAD2ENF>
      <inputfile path="${INPUTFILE}"/>
      <outputfile path="${OUTPUTNAME}_bf.enf"/>
      <parameterfile path="${PARAMETERFILE}"/>
      <productcode id="${PRODUCTID}"/>
      <workdir path="${WORKDIR}"/>
      <logfile path="${OUTPUTNAME}.log"/>
      <xmllogfile path="${OUTPUTNAME}.xml"/>
    </CAD2ENF>

    <!-- 属性マッピングシナリオ -->
    <ATTREDIT>
      <inputfile path="${OUTPUTNAME}_bf.enf"/> <!-- 入力ファイル -->
      <outputfile path="${OUTPUTNAME}.enf"/> <!-- 出力ファイル -->
      <mapfile path="mappingfile.xml"/> <!-- マッピングファイル -->
      <bomfile path="bom.xml"/> <!-- 属性情報出力ファイル -->
      <workdir path="${WORKDIR}"/> <!-- 作業ディレクトリ -->
      <logfile path="${OUTPUTNAME}.log"/> <!-- ログファイル -->
    </ATTREDIT>
  </Scenario>
</ScenarioList>
```

II. 属性のマッピング

下記の3通りの属性編集が可能です。それぞれの操作を、“Rule”というタグで定義して、Attribute Editorを実行します。複数のRuleを同時に定義することも可能です。

1. 置換 (type="create")

変換元(From)から変換先(To)へ属性の値を置換します。

- 書式

変換元(From)と変換先(To)の属性のkeyとtypeをひとつのRuleとします。

Typeは"system" または "user" のどちらかの文字列で指定します。

2. 削除 (type="delete")

指定された属性を削除します。

- 書式

削除する属性(Attribute)を列挙します。

3. 結合 (type="combine")

変換元のFromノードの中に結合したい要素を順に並べます。

要素として指定可能なものは属性(Attribute)、任意の文字列(String)、連番(SerialNumber)です。

連番とは、変換先の属性が作成されるたびに1ずつ増える数字です。(初期値は0)

- Attribute : 削除機能(delete)と同様に指定します。
- String : value属性で結合したい文字列を指定します。
- SerialNumber : digit属性で桁数が指定可能です。(1～20、省略可)
桁数に満たない数字の頭を0で埋めます。(4桁で12の場合は"0012")
変換先は置換(create)と同様に指定します。

各 Rule に target 属性を指定することで、操作の対象要素を指定することができます。"model", "assembly", "part"で指定します。指定がない場合には、すべての要素に対して適用されます。

- マッピングファイル例

```
<?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>
```

注意)

- Ruleより上の階層では、タグが異なる場合は無視されます。
- Rule内で書式が不正な場合、そのRuleは無視されます。
- 同内容のRuleが記述されている場合には、警告を出力します。
- typeとkeyが同じ属性を複数作成するRuleが記述されている場合には、警告を出力し、二つ目以降は無視されます。
- マッピングファイル中で以下の文字を使用する場合は、次の定義に従って記述する必要があります。

```
'&'→&amp; '>'→&gt; '<'→&lt; '"'→ &quot; "'"→&apos;
例) × <String value="Name&Number"/>
     ○ <String value="Name&amp;Number"/>
```



- ”Rule”タグに、”force”属性(on/off)を追加し、onを指定することで、属性値が空である場合も、属性編集が実行されます。

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

- System属性は、下記のリストに記述されたものに限定されます。それ以外のは無視されます。

System属性として認識可能なKeyの一覧

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

III. 属性の抽出

ENFに含まれる属性情報を、別ファイルとしてXML形式で書き出すことが可能です。出力ファイルは、シナリオXML中に<bomfile>として指定します。

- 出力されるBOMファイルの例
アセンブリ中のパート名、材質の一覧表となる。

```

<?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. 属性値の置換

ENFに含まれる属性の値を一定のルールで置換することができます。パート、アセンブリ、モデルに付加されたシステム属性/ ユーザ属性が置換可能です。また、置換対象の属性値の指定に関しては、正規表現が使用可能です。

<ValueReplacementRules> タグ

本機能の設定は AttrEditタグ中の ValueReplacementRules タグの内側に記述します。複数の Rule の記述も可能です。

<Rule> タグ

属性として以下のオプションを設定することができます。(任意)

- target: "part", "assembly", "model" を指定します。指定されたもののみが置換対象となります。無指定の場合はすべて置換します。

- **regexp** : "on", "off" を指定します。on の場合は置換する文字列の指定を正規表現として扱います。無指定の場合は off として扱います。
- **Rule タグの内部**に対象のAttributeと置換前後の文字列を指定します。対象のAttributeとtypeが同一のルールが複数存在する場合は、二つ目以降は無視してログに警告を出力します。

<Attribute> タグ

以下の属性により変換の対象を指定します。(必須)

- **key** : Attributeのキーを指定します。
- **type** : "system", "user" を指定し、システム属性、ユーザ属性を区別します。

<Value> タグ

置換前後のAttributeの値を指定します。(必須)

- **from** : 検索する文字列を指定します。
- **to** : 置換後の文字列を指定します。

Attributeタグに続けて記述します。複数設定することが可能です。

検索する文字列に一致する部分は全て置換を行います。上に書かれているルールから順に一致するかどうかを確認し、一度置換が行われると、それ以降に一致する条件があっても無視されます。

- 設定ファイルの例

```
<?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>
```

システム属性であるMaterial属性値に関して、"Steel"を"Metal"に、"Copper"を"Bronze"に置換します。また、AAAで始まるすべてのName属性値をABCに置換します。

V. 属性値の追加

ENFに属性を追加することができます。パート、アセンブリ、モデルに付加されたシステム属性とユーザ属性が対象となります。

<ValueAdditionRules> タグ

本機能の設定は AttrEdit タグ中の ValueAdditionRules タグの内側に記述します。複数の Rule を記述可能です。

属性として以下のオプションを設定可能です。(任意)

- **overwrite** : 上書き属性を "on", "off" で指定します。on の場合既に同名の属性が存在しても上書きします。無指定の場合はonとして扱います。システム属性とユーザ属性はキーが同一でも別のものとして扱います。

<Rule> タグ

属性として以下のオプションを設定可能です。(任意)

- **target** : "part", "assembly", "model" を指定。指定されたもののみを、属性追加の対象とします。無指定の場合はすべて置換します。Rule タグの内部に対象の追加対象の名前と追加するAttributeとその値を指定します。

<TargetName> タグ

属性追加対象のパート名(ENFのBody名)・アセンブリ名をvalue属性として指定します。名前が完全一致した場合のみ属性の追加を行います。このタグが存在しない場合は全てが追加対象となります。また、Ruleでtarget="model"が指定されている場合はこのタグの有無、value属性の内容によらず、モデルにAttributeを追加します。

属性として以下のオプションを指定可能です。

- **regexp** : 正規表現のフラグとして、"on", "off" を指定します。on の場合は対象の指定を正規表現として扱い、無指定の場合は off として扱います。

<Attribute> タグ

以下の属性により変換の対象を指定します。(必須)

- **key** : Attribute のキーを指定します。
- **type** : "system", "user" を指定します。

<Value> タグ

追加する Attribute の値を、タグの内部に指定します。(必須)

- 例)

```
<?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>
```

Arm200-101で始まるパートに対して、マテリアル属性を付加します。
 Arm200-asm-で始まるアセンブリに対して、Maker属性を付加します。
 Modelに対して、Designer属性を付加します。

VI. インスタンス名の編集

ENFのインスタンス名を一定のルールで編集することが可能です。

<TargetSpecificRules> タグ

本機能の設定は AttrEdit タグ中の TargetSpecificRulesタグの内側に記述します。

<Rule> タグ

属性として以下のオプションを設定します。(必須)

type="combine"

target="instance"

target: "instance"はインスタンス名の編集時にのみ有効です

<From> タグ

Fromノードの中に、結合したい要素を順に並べます。

要素として指定可能なものは、属性(Attribute)、任意の文字列(String)、連番(SerialNumber)です。

<String> タグ

value属性で結合したい文字列を指定します。

<SerialNumber> タグ

digit属性で桁数を指定可能です。(1～20、省略可)

桁数に満たない数字の頭を0で埋めます。(4桁で12の場合は"0012")

モデル全体のインスタンス名に通し番号がつきます。

<Attribute> タグ

以下の属性により対象を指定します。(必須)

- source : "Parent ", "Child "を指定します。それぞれ、該当のインスタンス直上のアセンブリ、該当のインスタンス直下のパートまたはアセンブリ (実体) を意味します。
- key : sourceで指定された要素のAttributeのキーを指定します。
- type : "system", "user" を指定しシステム属性、ユーザ属性を区別します。

<To> タグ

属性として以下のオプションを設定します。(必須)

key="Name"

- 設定ファイルの例
親のアセンブリ名と子のパートまたはアセンブリ名を"_"で結合し追番をつける

```
<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

ENFのアセンブリ構造の編集を行うコンポーネントです。下記の操作が可能です。

- [パート/アセンブリの削除](#)
- [アセンブリ構造の変更](#)
- [パート/アセンブリの抽出](#)

I. Assembly Editorの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

- シナリオ例 (パート/アセンブリの削除)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<ScenarioList>
  <Scenario>
    <!-- CADからENFに変換する -->
    <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>

    <!--要素削除シナリオ -->
    <ASMEDIT>
      <inputfile path="\${OUTPUTNAME}_a.enf"/> <!-- 入力ファイル -->
      <outputfile path="\${OUTPUTNAME}_b.enf"/> <!-- 出力ファイル -->
      <parameterfile path="\${PARAMETERFILE}"/> <!--パラメータファイル-->
      <namelist path="deleteList.txt"/> <!--リストファイル -->
      <workdir path="\${WORKDIR}"/> <!-- 作業ディレクトリ -->
      <logfile path="\${OUTPUTNAME}_b.log"/> <!-- ログファイル -->
    </ASMEDIT>
  </Scenario>
</ScenarioList>
```

- シナリオ例 (パート/アセンブリの抽出)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<ScenarioList>
  <Scenario>
    <ASMEDIT>
      <inputfile path="\${INPUTNAME}.enf"/> <!-- 入力ファイル -->
      <outputfile path="\${OUTPUTNAME}.enf"/> <!-- 出力ファイル -->
      <parameterfile path="\${PARAMETERFILE}"/> <!--パラメータファイル-->
      <extraction_target path="target.xml"/> <!-- 抽出対象リスト -->
      <extraction_outputdir path="\${OUTPUTPATH}\extraction"/>
      <!-- 抽出結果ENF出力先フォルダ -->
      <workdir path="\${WORKDIR}"/> <!-- 作業ディレクトリ -->
      <logfile path="\${INPUTNAME}_extraction.log"/> <!-- ログファイル -->
    </ASMEDIT>
  </Scenario>
</ScenarioList>
```

II. パート/アセンブリの削除

パートまたはアセンブリの削除対象リストをシナリオに指定して、削除を行います。設定パラメータについては、パラメータマニュアルをご参照ください。

- 削除対象リストの書式
 - 削除するパート・アセンブリ名を改行区切りで列挙したテキストです。正規表現指定パラメータ(UseRegexpInNameList)の値が0以外の場合は、リスト中にある名前は正規表現とみなされます。
 - パート削除パラメータ (DeleteSpecifiedPart) の値が1の場合は、リスト中にある名前のいずれかに“完全に一致する”パートが削除されます。また、アセンブリ削除パラメータ (DeleteSpecifiedAssembly) の値が1の場合は、同様にリスト中の名前に“完全に一致する”アセンブリが削除されます。
- 仕様説明
 - 同一名称の要素があった場合は全て削除します。
 - 削除したことによって親のアセンブリが空になった場合でも、親のアセンブリは削除されません。
 - アセンブリを削除する場合、その子要素を含めて全て削除されますが、別のアセンブリに共有されているものは削除されません。対象のインスタンスのみが削除されます。
 - 本機能実行時に削除対象リストが未指定、存在しない、読み込み禁止の場合には、その後の処理は実行されずプログラムが終了します。ログには“Failed to open list”と出力されます。
 - 削除対象が存在しない場合、その指定がなかったものとして処理が続行されます。ログに“Component XXX was not found in the model”と出力されます。

III. アセンブリの構造の編集

ENFのアセンブリ構造に関する下記の操作が可能です。詳しい使用方法是パラメータマニュアルをご参照ください。

- アセンブリをパートにまとめる
- 共有パート/アセンブリを複製する
- トップアセンブリを作成する
- 空のパート/アセンブリを削除する
- ボリューム要素をパートに変更する

IV. パート/アセンブリの抽出

パートまたはアセンブリの抽出を行う機能です。抽出対象リストと抽出結果出力フォルダをシナリオXMLで指定します。設定パラメータについては、パラメータマニュアルをご参照ください。

- 抽出対象リストの書式
抽出するパート・アセンブリ名を列挙したXMLファイル(*.xml)です。正規表現指定パラメータ (ExtractByRegularExpression)の値が0以外の場合は、リスト中にある名前、パス名は正規表現と

みなされます。

<AssemblyExtractionList> タグ

本機能の設定は AssemblyExtractionListタグ中に記述します。

内側に複数のAssemblyExtractionRuleタグを記述することで、抽出対象を指定します。

<AssemblyExtractionRule> タグ

本機能の設定は AssemblyExtractionListタグ中の AssemblyExtractionRuleタグの内側に記述します。

AssemblyExtractionRuleタグを記述することで抽出対象を指定します。

AssemblyExtractionRuleタグはそれぞれ一つの抽出対象に対応しており、

AssemblyExtractionRuleタグは抽出対象の数だけ指定します。

AssemblyExtractionRuleタグの内側には、type属性 / keep_position属性を記述することで、抽出の方法を指定することができます。

- type : (必須)

抽出対象の探索方法を指定します。

「単一名指定」「名前パス指定」「インスタンスパス指定」の3通りがあり、それぞれ「name」、「namePath」、「instancePath」で指定します。

- keep_position : (任意)

"yes", "no" を指定します。

yes の場合は上位アセンブリ付で、noの場合は上位アセンブリなしで抽出します。無指定の場合は no として扱われます。

注) keep_position属性は、type=name(単一名指定)でのみ指定可能です。

AssemblyExtractionRuleタグの内側には、抽出対象を指定します。

下記の3通りの指定が可能です。

1. 単一名指定(type="name")

Assemblyタグ・Partタグを指定します。

指定されたアセンブリまたはパートが抽出対象となります。

2. 名前パス指定(type="namePath")

Assemblyタグ内にPartタグが入れ子になる形で指定します。

アセンブリツリーのトップから目的のパートまでの階層構造を、アセンブリ名およびパート名で記述します。

3. インスタンスパス指定(type="instancePath")

Instanceタグ内にInstanceタグが入れ子になる形で指定します。

アセンブリツリーのトップから目的のパートまでの階層構造を、インスタンス名で記述します。

<Assembly> タグ

抽出対象のアセンブリ名を記述します。

Assemblyタグの内側には以下の属性を記述します。

- name : (必須)

抽出対象名を指定します。

ExtractByRegularExpressionが0以外の場合には、正規表現を指定可能です。

<Part> タグ

抽出対象のパート名を記述します。

Partタグの内側には以下の属性を記述します。

- name：(必須)

抽出対象名を指定します。

ExtractByRegularExpressionが0以外の場合には、正規表現を指定可能です。

<Instance> タグ

抽出対象のインスタンス名を記述します。

Instanceタグの内側には以下の属性を記述します。

- name：(必須)

抽出対象名を指定します。

ExtractByRegularExpressionが0以外の場合には、正規表現を指定可能です。



- タグ名、属性名について、文字列の大文字小文字は区別します。
- 属性を記述する順序は任意です。
- 値はXMLの書式に準じて記述する必要があり、XMLファイル中で以下の文字を使用する場合は、次の定義に従って記述する必要があります。

'&' → & '>' → > '<' → < '"' → " "'" → '

- 抽出対象リストの例

```
<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>
```

- 抽出動作の仕様

- 指定可能なパターン

パート・アセンブリ名、パート・アセンブリ名パス、インスタンス名パスの3パターンで指定可能です。

- 抽出対象

抽出対象リストで指定した名前に一致したアセンブリ・パートを抽出対象とします

- 抽出結果

抽出結果はENFに書き出されます。出力されるENFの名前には抽出対象要素の名前が使用されます。

- Windowsファイル名使用禁止文字が含まれる場合には、使用禁止文字を"_"（アンダースコア）に置換した名前が使用されます。
- パート・アセンブリ名パス、インスタンス名パスを指定した場合には、各パート・アセンブリ名および各インスタンス名を"_"（アンダースコア）でつなげた名前が使用されます。
- keep_position属性の値(yes/no)以外が同じルールでの、2つの抽出結果の出力ENF名は同じとなるため、後に記述したルールでの抽出結果で出力ENFが上書きされます。そのため、keep_position属性の値(yes/no)以外が同じルールでの抽出結果を別々に出力したい場合には、Assembly Editorを2回に分けて実行する必要があります。
- <同名で実体異なる抽出対象が複数ある場合>
別々のENFとして抽出されます。それぞれのENF名は抽出対象の名前に追い番がついた名前となります。
- <抽出対象が共有されている場合>
全て同一のENFに抽出されます。ただし、keep_position=noの場合には、抽出対象が共有されていても一つの実体としてENFに抽出されます。

- 抽出動作例

以下のアセンブリを、上記の抽出対象リストを使用して抽出した例を示します。

- 元アセンブリ構造

-括弧内はインスタンス名

-SubAsm1が共有されています

```
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

出力ファイル名 : Part1.enf

Part1のみが含まれるENFが出力されます

2. Rule 2

出力ファイル名 : Part2.enf

Part2が上位アセンブリ付きで出力されます。出力されるENFの構造は下記のようになります。

```

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

```

3. Rule 3

出力ファイル名 : SubAsm2.enf

元のサブアセンブリであるSubAsm2がトップアセンブリで、下位にPart1, Part2を持つENFが出力されます。出力されるENFの構造は下記のとおりです。

```

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

```

4. Rule 4

出力ファイル名 : TopAsm_Part3.enf

Part3が上位アセンブリ付きで出力されます。出力されるENFの構造は下記のとおりです。

```

TopAsm
|-----Part3 (Part3_instance1)

```

5. Rule 5

出力ファイル名 :

TopAsm_SubAsm1_SubAsm2_Part1_1.enf

TopAsm_SubAsm1_SubAsm2_Part1_2.enf

Part1が上位アセンブリつきで、2ファイル出力されます。出力されるENFの構造は下記のとおりです。


```

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

```

```

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

```

6. Rule 6

出力ファイル名：

SubAsm1_instance1_SubAsm2_instance1_Part2_insancet1.enf

Part2が上位アセンブリ付きで出力されます。出力されるENFの構造は下記のとおりです。

```

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

```

3.3.6. Interference Checker

ENFのアセンブリに対して、干渉チェックを行うコンポーネントです。下記の操作が可能です。

- 部品間の干渉チェック
- 部品間の接触チェック

I. Interference Checkerの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

- シナリオ例

```

<?xml version="1.0" encoding="Shift_JIS"?>
<ScenarioList>
  <Scenario>
    <!-- CADからENFに変換する -->
    <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>

    <!--干渉チェックシナリオ -->
    <INTRCHK>
      <inputfile path="${OUTPUTNAME}_a.enf"/> <!-- 入力ファイル -->
      <csvfile path="${OUTPUTNAME}_a.csv"/> <!-- 検証結果ファイル -->
      <parameterfile path="${PARAMETERFILE}"/> <!-- パラメータファイル -->
      <workdir path="${WORKDIR}"/> <!-- 作業ディレクトリ -->
      <logfile path="${OUTPUTNAME}_b.log"/> <!-- ログファイル -->
    </INTRCHK >
  </Scenario>
</ScenarioList>

```

干渉チェックと接触チェックの切り替えはパラメータファイルの設定で行います。

[INTRCHECK]

CheckMode=1

0:干渉チェック、1:接触チェック [デフォルト:0]

II. 干渉チェック

パート間の干渉チェックを行い、その結果をCSVファイルへ書き出すことが可能です。トレランスや検証対象などの条件、出力する情報の選択などがパラメータにより制御可能です。詳細はパラメータマニュアルをご参照ください。

また、干渉チェック後に出力されるCSVファイルと、もとのENFファイルをModel Viewerに読み込むことにより、干渉位置を画面上で確認することが可能です。詳しいご利用方法はModel Viewerのヘルプをご参照ください。

- 出力ファイル形式
出力ファイルはCSV形式で以下のように出力されます。

パラメータ	←	・パラメータとその値を表示
トレランス,0.01	←	・干渉の許容値(mm)
深さ計測点の間隔,0.23	←	・自動の場合のみ計算された値が表示される
部品1,部品2,干渉部分の体積,深さ	←	・1行改行
部品A,部品B,50.23,10.12	←	・部品1と部品2の干渉部品の体積(mm ³)、最短距離、重心、食い込みの深さ(mm)のうち、パラメータで出力指定されているものを表示
部品A,部品C,0,0		・干渉していないが、部品間距離が指定トレランス以内のものは、干渉部品の体積を0で表示。
部品B,部品C,26.45,3.0		・部品間距離が指定トレランス以上のものは表示しない。
部品C,部品D,3.54,4.56		
部品C,部品E,15.26,2.0	←	
...		

III. 接触チェック

パート間の接触チェックを行い、その結果をCSVファイルに書き出すことが可能です。パラメータClearanceで指定した距離の中に、パラメータAreaToleranceで指定した面積以上の重複が検出された場合に接触と判定します。その他にトレランスや検証対象などの条件、出力する情報の選択などがパラメータにより制御可能です。詳細はパラメータマニュアルをご参照ください。

- 出力ファイル形式

出力ファイルはCSV形式で以下のように出力されます。

パラメータ	←	・パラメータとその値を表示
トレランス,0.01,0.1	←	・距離トレランス、面積トレランス(mm)
部品1,部品2,接触部分の面積,面積1,面積2	←	・1行改行
部品A,部品B,10.23,20,30	←	・部品1と部品2の接触部分の面積(mm ²)、接触フェースの面積1、面積2を表示
部品A,部品C,0,0,0		
部品B,部品C,26.45,30.5,60.8		
部品C,部品D,3.54,20.56,22		
部品C,部品E,1.26,8.0,10.2		
...		

3.3.7. Polygon Optimizer

指定されたファイルのポリゴンデータを自動で最適化するコンポーネントです。ポリゴン最適化の機能として下記が実行可能です。

自動修正機能	ポリゴンを自動的に最適化します
平滑化機能	ノイズ等の含まれるポリゴンを平滑化し、滑らかな形状にします
ピン角化機能	角がぼやけているポリゴンを、ピン角を強調したポリゴンにします

リメッシュ機能	ポリゴンデータに対してメッシュを再作成します
ラッピング機能	バラバラに分かれているメッシュを統合して閉じたポリゴンモデルを作成します
オフセット機能	ポリゴンのオフセット形状を作成します
穴埋め機能	ポリゴンの穴（フリーエッジ部分）を埋めます
スティッチ機能	ポリゴンデータに含まれるポリゴン間の隙間(フリーエッジ)を縫い合わせます
シャープエッジの一括修正機能	シャープエッジ(連なるポリゴンの法線が、閾値以上折れている箇所)を修正します
シャープエッジの削除と穴埋め機能	シャープエッジ(連なるポリゴンの法線が、閾値以上折れている箇所)の削除と穴埋めを実行します
短いエッジの一括修正機能	長さが閾値以下のポリゴンエッジを修正します
長いエッジの一括修正機能	長さが閾値以上のポリゴンエッジを修正します
微小角のあるフェースの一括修正機能	閾値以下の角度をもつポリゴンフェースを修正します
微小シェルの一括修正機能	面積が閾値以下の微小なシェルを修正します
微小ボリュームの一括修正機能	体積が閾値以下の微小なボリュームを修正します
内部空洞の一括修正機能	内部空洞を修正します
簡略化機能	大規模なポリゴンの軽量化を実行します

I. Polygon Optimizerの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

II. ポリゴン最適化項目の設定

パラメータファイルにポリゴン最適化手順と項目を記述します。

注) 詳細は、パラメータマニュアルを参照してください。

• 例

```
Process_0=AutoHeal           //1番目 自動修正
ModifyPolygonSmallShell_0=1
ModifyPolygonSmallVolume_0=1
ModifyPolygonCavity_0=1
Process_1=Remesh             //2番目 リメッシュ
Process_2=Smooth              //3番目 平滑化
ProcessNumberMax=3           //
```

III. 制限

自動修正、ラッピング以外の機能は、不正位相(非多様体頂点、非多様体エッジ、表裏不整合、重複三角形)がある状態では実行できません。そのようなデータを使用する場合は、事前に自動修正を実行し

てください。なお、縮退三角形は各機能の実行前に自動的に削除されます。

オフセット機能は、周りの三角形の全てが極端に短いエッジあるいは極端に小さい内角を持つような頂点が存在する場合には実行できません。事前に短いエッジや微小角のあるフェースを修正するようにしてください。

3.3.8. ENF Editor

ENF編集を行うためのコンポーネントです。実行方法として2種類あります。

- 任意のスクリプト記述によるENF編集
- 標準搭載のスクリプトを用いて設定ファイルを指定することによるENF編集
(アセンブリ編集、属性編集)

3.3.8.1. 任意スクリプトの実行

ENFの情報を取得/編集できるENF Editor API for Rubyを使いスクリプトを自由に記述することでENFの情報の取得/編集をすることが可能です。例えば下記の操作が可能です。

- コンポーネントのユーザ属性/システム属性の追加・削除・マッピングを行う
- コンポーネント、ボリューム、フェースの要素数をカウントし、ログに書き出す
- ボリューム、フェース、エッジの色・表示非表示・レイヤを変更する
- ボリューム名にボリュームのレイヤ番号を付加する
- 特定のレイヤに存在するボリュームを非表示にする

I. 実行方法

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

- シナリオ例

```
<?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. スクリプトの記述方法

スクリプトの記述方法に関しては別冊の「ENF Editor Manual」 および 「ENF Editor API Reference Guide」を参照して下さい。

3.3.8.2. 標準搭載スクリプトの実行(アセンブリ編集)

標準搭載のアセンブリ編集スクリプトを用いて、アセンブリ構造の編集を行うことが可能です。パラメータ、削除対象リスト(namelist)、抽出対象リスト(extraction_target)など固定のフォーマットの設定ファイルを指定することで下記の操作が可能です。

- [パート/アセンブリの削除](#)
- [アセンブリ構造の編集](#)
- [パート/アセンブリの抽出](#)

注) 本スクリプトはEX5.x 以前のAssembly Editorコンポーネントと同じ動作をするスクリプトです。削除対象リスト、抽出対象リストは同一のファイルを再利用することが可能です。

I. 実行方法

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

- シナリオ例 (パート/アセンブリの削除)

```
<?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>
```

- シナリオ例 (パート/アセンブリの抽出)

```
<?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. パート/アセンブリの削除

パートまたはアセンブリの削除対象リストをシナリオに指定して、削除を行います。設定パラメータについては、パラメータマニュアルをご参照ください。

• 削除対象リストの書式

- 削除するパート・アセンブリ名を改行区切りで列挙したテキストです。正規表現指定パラメータ(UseRegexpInNameList)の値が0以外の場合は、リスト中にある名前は正規表現とみなされます。
- パート削除パラメータ (DeleteSpecifiedPart) の値が1の場合は、リスト中にある名前のいずれかに“完全に一致する”パートが削除されます。また、アセンブリ削除パラメータ (DeleteSpecifiedAssembly) の値が1の場合は、同様にリスト中の名前に“完全に一致する”アセンブリが削除されます。

• 仕様説明

- 同一名称の要素があった場合は全て削除します。
- 削除したことによって親のアセンブリが空になった場合でも、親のアセンブリは削除されません。
- アセンブリを削除する場合、その子要素を含めて全て削除されますが、別のアセンブリに共有されているものは削除されません。対象のインスタンスのみが削除されます。
- 本機能実行時に削除対象リストが未指定、存在しない、読み込み禁止の場合には、その後の処理は実行されずプログラムが終了します。ログには“Failed to open list”と出力されます。
- 削除対象が存在しない場合、その指定がなかったものとして処理が続行されます。ログに“Component XXX was not found in the model”と出力されます。

III. アセンブリの構造の編集

ENFのアセンブリ構造に関する下記の操作が可能です。詳しい使用方法是パラメータマニュアルをご参照ください。

- アセンブリをパートにまとめる
- 共有パート/アセンブリを複製する
- トップアセンブリを作成する
- 空のパート/アセンブリを削除する
- ボリューム要素をパートに変更する

IV. パート/アセンブリの抽出

パートまたはアセンブリの抽出を行う機能です。抽出対象リストと抽出結果出力フォルダをシナリオXMLで指定します。設定パラメータについては、パラメータマニュアルをご参照ください。

- 抽出対象リストの書式

抽出するパート・アセンブリ名を列挙したXMLファイル(*.xml)です。正規表現指定パラメータ (ExtractByRegularExpression)の値が0以外の場合は、リスト中にある名前、パス名は正規表現とみなされます。

<AssemblyExtractionList> タグ

本機能の設定は AssemblyExtractionListタグ中に記述します。

内側に複数の AssemblyExtractionRuleタグを記述することで、抽出対象を指定します。

<AssemblyExtractionRule> タグ

本機能の設定は AssemblyExtractionListタグ中の AssemblyExtractionRuleタグの内側に記述します。

AssemblyExtractionRuleタグを記述することで抽出対象を指定します。

AssemblyExtractionRuleタグはそれぞれ一つの抽出対象に対応しており、AssemblyExtractionRuleタグは抽出対象の数だけ指定します。

AssemblyExtractionRuleタグの内側には、type属性 / keep_position属性を記述することで、抽出の方法を指定できます。

- type : (必須)

抽出対象の探索方法を指定します。

「単一名指定」「名前パス指定」「インスタンスパス指定」の3通りがあり、それぞれ「name」, 「namePath」, 「instancePath」で指定します。

- keep_position : (任意)

"yes", "no" を指定します。

yesの場合は上位アセンブリ付で、noの場合は上位アセンブリなしで抽出します。無指定の場合はnoとして扱われます。

注): keep_position属性は、type=name(単一名指定)でのみ指定可能です。

AssemblyExtractionRuleタグの内側には、抽出対象を指定します。

下記の3通りの指定が可能です。

- a. 単一名指定(type="name")
Assemblyタグ・Partタグを指定します。
指定されたアセンブリまたはパートが抽出対象となります。
- b. 名前パス指定(type="namePath")
Assemblyタグ内にPartタグが入れ子になる形で指定します。
アセンブリツリーのトップから目的のパートまでの階層構造を、アセンブリ名およびパート名で記述します。
- c. インスタンスパス指定(type="instancePath")
Instanceタグ内にInstanceタグが入れ子になる形で指定します。
アセンブリツリーのトップから目的のパートまでの階層構造を、インスタンス名で記述します。

<Assembly> タグ

抽出対象のアセンブリ名を記述します。

Assemblyタグの内側には以下の属性を記述します。

- name : (必須)
抽出対象名を指定します。
ExtractByRegularExpressionが0以外の場合には、正規表現を指定可能です。

<Part> タグ

抽出対象のパート名を記述します。

Partタグの内側には以下の属性を記述します。

- name : (必須)
抽出対象名を指定します。
ExtractByRegularExpressionが0以外の場合には、正規表現を指定可能です。

<Instance> タグ

抽出対象のインスタンス名を記述します。

Instanceタグの内側には以下の属性を記述します。

- name : (必須)
抽出対象名を指定します。
ExtractByRegularExpressionが0以外の場合には、正規表現を指定可能です。



- タグ名、属性名について、文字列の大文字小文字は区別します。
- 属性を記述する順序は任意です。
値はXMLの書式に準じて記述する必要があり、XMLファイル中で以下の文字を使用する場合は、次の定義に従って記述する必要があります。

‘&’→ &, ‘>’→ >, ‘<’→ <, ‘”’→ ", ‘’’→ '

- 抽出対象リストの例

```
<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>
```

- 抽出動作の仕様

- 指定可能なパターン

パート・アセンブリ名、パート・アセンブリ名パス、インスタンス名パスの3パターンで指定可能です。

- 抽出対象

抽出対象リストで指定した名前に一致したアセンブリ・パートを抽出対象とします

- 抽出結果

抽出結果はENFに書き出されます。出力されるENFの名前には抽出対象要素の名前が使用されます。

- Windowsファイル名使用禁止文字が含まれる場合には、使用禁止文字を"_"（アンダースコア）に置換した名前が使用されます。
- パート・アセンブリ名パス、インスタンス名パスを指定した場合には、各パート・アセンブリ名および各インスタンス名を"_"（アンダースコア）でつなげた名前が使用されます。
- keep_position属性の値(yes/no)以外が同じルールでの、2つの抽出結果の出力ENF名は同じとなるため、後に記述したルールでの抽出結果で出力ENFが上書きされます。そのため、keep_position属性の値(yes/no)以外が同じルールでの抽出結果を別々に出力したい場合には、Assembly Editorを2回に分けて実行する必要があります。
- <同名で実体異なる抽出対象が複数ある場合>
別々のENFとして抽出されます。それぞれのENF名は抽出対象の名前に追いついた名前となります。
- <抽出対象が共有されている場合>
全て同一のENFに抽出されます。ただし、keep_position=noの場合には、抽出対象が共有されていても一つの実体としてENFに抽出されます。

- 抽出動作例

以下のアセンブリを、上記の抽出対象リストを使用して抽出した例を示します。

- 元アセンブリ構造

-括弧内はインスタンス名

-SubAsm1が共有されています

```
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

出力ファイル名 : Part1.enf

Part1のみが含まれるENFが出力されます

2. Rule 2

出力ファイル名 : Part2.enf

Part2が上位アセンブリ付きで出力されます。出力されるENFの構造は下記のようになります。

```

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

```

3. Rule 3

出力ファイル名 : SubAsm2.enf

元のサブアセンブリであるSubAsm2がトップアセンブリで、下位にPart1, Part2を持つENFが出力されます。出力されるENFの構造は下記のとおりです。

```

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

```

4. Rule 4

出力ファイル名 : TopAsm_Part3.enf

Part3が上位アセンブリ付きで出力されます。出力されるENFの構造は下記のとおりです。

```

TopAsm
|-----Part3 (Part3_instance1)

```

5. Rule 5

出力ファイル名1 :

TopAsm_SubAsm1_SubAsm2_Part1_1.enf

TopAsm_SubAsm1_SubAsm2_Part1_2.enf

Part1が上位アセンブリつきで、2ファイル出力されます。出力されるENFの構造は下記のとおりです。

```

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

```

```

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

```

6. Rule 6

出力ファイル名 : SubAsm1_instance1_SubAsm2_instance1_Part2_insancet1.enf

Part2が上位アセンブリ付きで出力されます。出力されるENFの構造は下記のとおりです。

```

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

```

3.3.8.3. 標準搭載スクリプトの実行(属性編集)

標準搭載の属性編集スクリプトを用いて、属性情報の操作を行うことが可能です。

マップファイル(mapfile)、属性リストファイル(bomfile)などを指定することで下記の操作を行うことが可能です。

- [属性のマッピング](#)
- [属性の抽出](#)
- [属性値の置換](#)
- [属性値の追加](#)
- [インスタンス名の編集](#)

注) 本スクリプトはEX5.x 以前のAttribute Editorコンポーネントと同じ動作をするスクリプトです。

マップファイルは同一のファイルを再利用することが可能です。

I. 実行方法

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

```
<?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. 属性のマッピング

下記の3通りの属性編集が可能です。それぞれの操作を、“Rule”というタグで定義して、Attribute Editorを実行します。複数のRuleを同時に定義することも可能です。

1. 置換 (type="create")

変換元(From)から変換先(To)へ属性の値を置換します。

書式

- 変換元(From)と変換先(To)の属性のkeyとtypeを、ひとつのRuleとします。
- Typeは"system" または "user" のどちらかの文字列で指定します。

2. 削除 (type="delete")

指定された属性を削除します。

書式

- 削除する属性(Attribute)を列挙します。

3. 結合 (type="combine")

変換元のFromノードの中に結合したい要素を順に並べます。

要素として指定可能なものは、属性(Attribute)・任意の文字列(String)・連番(SerialNumber)です。

連番とは、変換先の属性が作成されるたびに、1ずつ増える数字です。(初期値は0)

- Attribute : 削除機能(delete)と同様に指定します。
- String : value属性で結合したい文字列を指定します。
- SerialNumber : digit属性で桁数が指定可能です。(1~20、省略可)
桁数に満たない数字の頭を0で埋めます。(4桁で12の場合は"0012")
変換先は置換(create)と同様に指定します。

各 Rule に target 属性を指定することで、操作の対象要素を指定することができます。"model", "assembly", "part"で指定します。

指定がない場合には、すべての要素に対して適用されます。

- マッピングファイル例

```
<?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>
```



- Ruleより上の階層では、タグが異なる場合は無視されます。
- Rule内で書式が不正な場合、そのRuleは無視されます。
- 同内容のRuleが記述されている場合には、警告を出力します。
- typeとkeyが同じ属性を複数作成するRuleが記述されている場合には、警告を出力し、二つ目以降は無視されます。

マッピングファイル中で以下の文字を使用する場合は、次の定義に従って記述する必要があります。

‘&’→ & ‘>’→ > ‘<’→ < ‘”’→ " ‘’’→ '

例) × <String value="Name&Number"/>

○ <String value="Name&Number"/>



- ”Rule”タグに”force”属性(on/off)を追加しonを指定することで、属性値が空である場合も属性編集が実行されます。

例) <Rule type=”combine” force=”on”>

- System属性は、下記のリストに記述されたものに限定されます。それ以外のは無視されます。

System属性として認識可能なKeyの一覧

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

III. 属性の抽出

ENFに含まれる属性情報を、別ファイルとしてXML形式で書き出すことが可能です。出力ファイルは、シナリオXML中に<bomfile>として指定します。

- 出力されるBOMファイルの例。 アセンブリ中のパート名、材質の一覧表となる。

```
<?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. 属性値の置換

ENFに含まれる属性の値を一定のルールで置換することができます。パート、アセンブリ、モデルに付加されたシステム属性/ ユーザ属性が置換可能です。また、置換対象の属性値の指定に関しては、正規表現が使用可能です。

<ValueReplacementRules> タグ

本機能の設定は AttrEdit タグ中の ValueReplacementRules タグの内側に記述します。複数の Rule の記述も可能です。

<Rule> タグ

属性として以下のオプションを設定することができます。(任意)

target: "part", "assembly", "model" を指定します。指定されたもののみが置換対象となります。無指定の場合はすべて置換します。

regexp: "on", "off" を指定。on の場合は置換する文字列の指定を正規表現として扱います。無指

定の場合は off として扱います。

Rule タグの内部に対象のAttributeと置換前後の文字列を指定します。

対象のAttributeとtypeが同一のルールが複数存在する場合は、二つ目以降は無視してログに警告を出力します。

<Attribute> タグ

以下の属性により変換の対象を指定します。(必須)

key : Attribute のキーを指定します。

type : "system", "user"を指定し、システム属性、ユーザ属性を区別します。

<Value> タグ

置換前後の Attribute の値を指定します。(必須)

from : 検索する文字列を指定します。

to : 置換後の文字列を指定します。

Attributeタグに続けて記述します。複数設定することが可能です。

検索する文字列に一致する部分は全て置換を行います。上に書かれているルールから順に一致するかどうかを確認し、一度置換が行われると、それ以降に一致する条件があっても無視されます。

- 設定ファイルの例

```
<?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>
```

システム属性であるMaterial属性値に関して、"Steel"を"Metal"に、"Copper"を"Bronze"に置換します。また、AAAで始まるすべてのName属性値をABCに置換します。

V. 属性値の追加

ENFに属性を追加することができます。パート、アセンブリ、モデルに付加されたシステム属性とユーザ属性が対象となります。

<ValueAdditionRules> タグ

本機能の設定は AttrEdit タグ中の ValueAdditionRules タグの内側に記述します。複数の Rule を

記述可能です。

属性として以下のオプションを設定可能です。(任意)

overwrite : 上書き属性を "on", "off" で指定します。on の場合既に同名の属性が存在しても上書きします。無指定の場合はonとして扱います。システム属性とユーザ属性はキーが同一でも別のものとして扱います。

<Rule> タグ

属性として以下のオプションを設定可能です。(任意)

target : "part", "assembly", "model" を指定。指定されたもののみを、属性追加の対象とします。無指定の場合はすべて置換します。Ruleタグの内部に追加対象の名前と追加するAttributeとその値を指定します。

<TargetName> タグ

属性追加対象のパート名(ENFのBody名)・アセンブリ名をvalue属性として指定します。名前が完全一致した場合のみ属性の追加を行います。このタグが存在しない場合は全てが追加対象となります。また、Ruleでtarget="model"が指定されている場合はこのタグの有無、value属性の内容によらず、モデルにAttributeを追加します。

属性として以下のオプションを指定可能です。

regexp : 正規表現のフラグとして、"on", "off"を指定します。onの場合は対象の指定を正規表現として扱い、無指定の場合はoffとして扱います。

<Attribute> タグ

以下の属性により変換の対象を指定します。(必須)

key : Attributeのキーを指定します。

type : "system", "user"を指定します。

<Value> タグ

追加する Attribute の値を、タグの内部に指定します。(必須)

- 例

```
<?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>
```

Arm200-101で始まるパートに対して、Material属性を付加します。

Arm200-asm-で始まるアセンブリに対して、Maker属性を付加します。

Modelに対して、Designer属性を付加します。

VI. インスタンス名の編集

ENFのインスタンス名を一定のルールで編集することが可能です。

<TargetSpecificRules> タグ

本機能の設定はAttrEditタグ中のTargetSpecificRulesタグの内側に記述します。

<Rule> タグ

属性として以下のオプションを設定します。(必須)

type="combine"

target="instance"

target: "instance"はインスタンス名の編集時にのみ有効です。

<From> タグ

Fromノードの中に、結合したい要素を順に並べます。

要素として指定可能なものは属性(Attribute)、任意の文字列(String)、連番(SerialNumber)です。

<String> タグ

value属性で結合したい文字列を指定します。

<SerialNumber> タグ

digit属性で桁数を指定可能です。(1~20、省略可)

桁数に満たない数字の頭を0で埋めます。(4桁で12の場合は"0012")
 モデル全体のインスタンス名に通し番号がつきます。

<Attribute> タグ

以下の属性により対象を指定します。(必須)

source: "Parent", "Child"を指定します。それぞれ、該当のインスタンスの直上のアセンブリ、該当のインスタンス直下のパートまたはアセンブリ（実体）を意味します。

key: sourceで指定された要素のAttributeのキーを指定します。

type: "system", "user"を指定し、システム属性、ユーザ属性を区別します。

<To> タグ

属性として以下のオプションを設定します。(必須)

key="Name"

- 設定ファイルの例
 親のアセンブリ名と子のパートまたはアセンブリ名を"_"で結合し、追番をつける。

```
<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

3D PDF ENF Reader から出力する 3D PDF をカスタマイズする際に必要となる製品です。3D PDF に含まれるコンテンツやレイアウトを自由にカスタマイズすることが可能です。

I. 3D PDF Editorの実行

シナリオで実行します。「[2.4, “XMLシナリオ”](#)」を参照してください。

II. 注意点

カスタマイズを行う場合には、ASFALIS Data Package Studio を使用してください。詳細については、ASFALIS Data Package Studio のドキュメントをご参照ください。

3.4. Other

3.4.1. 3Dアノテーション変換

3Dアノテーション変換に対応している、各CADのバージョンについてはRequirementシート (Requirement.pdf)をご覧ください。

I. ライセンス

- 3DアノテーションをENF ReaderでCADに変換する際には、ENF Readerのライセンスの他に3Dアノテーション変換オプションのライセンスを必要とします。ライセンスコードについてはRequirementシート(Requirement.pdf)をご参照ください。(Writer側は標準機能)

II. 変換オプション

- 3Dアノテーションを変換するには、以下のオプションをパラメータファイルに記述します。

XConvertNote	注釈ノートを変換します。
XConvertGDT	幾何公差記号、データム、データムターゲットを変換します。
XConvertDimension	寸法を変換します。
XConvertSurfFinish	仕上げ記号を変換します。
XConvertSpotWeld XConvertLineWeld	溶接記号を変換します。
XConvertLocator	ロケータを変換します。
XConvertView	キャプチャ・ビューを変換します。
XConvertSection	断面を変換します。
XConvertHatching	ハッチングを変換します。
XConvert3DAnnotationAsPolyline	変換されるPMIに表示用ポリラインを付加します。
XConvertSymbol	シンボルを変換します。
XConvertGraphicDataPMI	中心線・領域を変換します。

- ENF Readerの3Dアノテーション変換において、Graphic情報をシンボルまたは単独線として変換する機能に対応しているAdapterがあります。変換方法は以下の各オプションで指定できます。

XConvertNote
 XConvertGDT
 XConvertDimension
 XConvertSurfFinish
 XConvertSpotWeld
 XConvertLineWeld
 XConvertLocator

III. 変換対象3Dアノテーション

1. CAD to ENF

◦ CATIA V5 to ENF

項目	CATIA V5 FTA
注記	Text / Flag Note
デーラム	Datum Feature
デーラムターゲット	Datum Target
幾何公差	Geometrical Tolerance
仕上げ記号	Roughness
寸法	Dimension
溶接	Spot Point / Weld Symbol

◦ 3DEXPERIENCE to ENF

項目	3D Tolerancing & Annotation
注記	Text / Flag Note
デーラム	Datum Feature
デーラムターゲット	Datum Target
幾何公差	Geometrical Tolerance
仕上げ記号	Roughness
寸法	Dimension
溶接	Weld Feature



すべての項目はポリラインとして変換されます。

◦ NX I-deas to ENF

項目	NX I-deas PMI
注記	Note
デーラム	Datum Feature Symbol
デーラムターゲット	Datum Target
幾何公差	Feature Control Frame
仕上げ記号	Surface Finish
寸法	Dimension

項目	NX I-deas PMI
溶接	Spot Weld / Line Weld
ロケータ	Locator

◦ Creo Parametric to ENF

項目	Creo Parametric
注記	Note
シンボル	Symbol
寸法 (*)	Dimension
幾何公差	Geometric Tolerance
仕上げ記号	Surface Finish
セットデータムタグ	Set Datum Tag
ノングラフィック	Non-Graphical

(*): 寸法の公差の表示有無は、変換に使用するCreo Parametricのコンフィギュレーション設定に従い変換されます。Adapterのコンフィギュレーション設定により、寸法の公差の表示有無を切り替えることもできます。

Creo Parametric Adapterコンフィギュレーションファイル:

旧アダプタ: <ASFALIS Components>\ProeExe\win\<Creo Version>\config.pro

新アダプタ: <ASFALIS Components> \CADFeature\Batch\Creo.x64\Parametric\<Creo Version>\config.pro

寸法の公差表示無し: tol_display=no

寸法の公差表示有り: tol_display=yes

初期設定では、以下のように”tol_display”の設定はコメントアウトされています。

!tol_display=yes

ただし、変換に使用するCreo Parametricのconfig.sup で ”tol_display” が設定されている場合、その設定が優先されます。

◦ NX to ENF

項目	NX
注記	Note / Drafting Note / Specialized Note
データム	Datum Feature Symbol / Drafting Datum Feature Symbol / Geometrical Tolerancing-Datum
データムターゲット	Datum Target / Geometrical Tolerancing-Datum Target

項目	NX
幾何公差	Feature Control Frame / Drafting Feature Control Frame / Geometrical Tolerancing-Tolerance
仕上げ記号	Surface Finish
寸法	Dimension / Drafting Dimension / Geometrical Tolerancing-Dimension
溶接	Spot Weld / Line Weld
カスタムシンボル	Custom Symbol
ロケータ	Locator
中心線	Center Line / Center Mark
領域	Region

◦ SOLIDWORKS to ENF

項目	SOLIDWORKS
注記	Note / Balloon
デーラム	Datum Feature
デーラムターゲット	Datum Target
幾何公差	Geometric Tolerance
寸法	Dimension / DimXpert
仕上げ記号	Surface Finish
溶接	Weld Symbol



PMI を変換するためには、新アダプタを使用する必要があります。旧アダプタで PMI を変換することはできません。

◦ JT to ENF

項目	JT
注記	Note / Coordinate Note / Balloon / Specialized Note
デーラム	Datum Feature Symbol
デーラムターゲット	Datum Target
幾何公差	Feature Control Frame
仕上げ記号	Surface Finish
寸法	Dimension

項目	JT
溶接	Spot Weld / Line Weld
ユーザ定義シンボル	User Defined Symbol
中心線	Center Line / Center Mark
領域	Region
ロケータ	Locator
切断平面シンボル	Cutting Plane Symbol

◦ STEP to ENF

項目	STEP
注記	Note / Label / General Note
データム	Datum
データムターゲット	Datum Target
幾何公差	Geometric Tolerance
仕上げ記号	Surface Roughness
寸法	Dimension
溶接	Weld Symbol

◦ CADmeister to ENF (standalone)

項目	CADmeister
注記	3D注記
寸法	参考寸法

2. ENF to CAD

◦ ENF to CATIA V5

本機能を使用するには、CATIA V5 ENF Reader 3Dアノテーション変換オプションが必要です。

項目	CATIA V5 FTA
注記	Text / Flag Note
データム	Datum Feature
データムターゲット	Datum Target
幾何公差	Geometrical Tolerance
仕上げ記号	Roughness

項目	CATIA V5 FTA
寸法	Dimension
溶接	Spot Point ^(*) / Weld Symbol
ロケータ	ポリラインとして変換

(*): 単独点として変換

◦ ENF to 3DEXPERIENCE

本機能を使用するには、3DEXPERIENCE Reader 3Dアノテーション変換オプションが必要です。

項目	3D Tolerancing & Annotation
注記	Text / Flag Note
データム	Datum Feature
データムターゲット	Datum Feature
幾何公差	Geometrical Tolerance
仕上げ記号	Roughness
寸法	Dimension
溶接	Weld Feature
ロケータ	Text



すべての項目はポリラインとして変換されます。

◦ ENF to NX I-deas

本機能を使用するには、NX I-deas ENF Reader 3Dアノテーション変換オプションが必要です。また、併せてNX I-deasのgeometric_tolライセンスが必要です。

項目	NX I-deas PMI
注記	Note
データム	Datum Feature Symbol
データムターゲット	Datum Target
幾何公差	Feature Control Frame
仕上げ記号	Surface Finish
寸法	Dimension
溶接	Spot Weld / Line Weld
ロケータ	Locator

◦ ENF to NX

本機能を使用するには、NX ENF Reader 3Dアノテーション変換オプションが必要です。
また、併せてNXのdrafting, geometric_tolライセンスが必要です。

項目	NX
注記	Note
デーラム	Datum Feature Symbol
デーラムターゲット	Datum Target
幾何公差	Feature Control Frame
寸法	Dimension
仕上げ記号	Surface Finish
カスタムシンボル	Custom Symbol



Annotation機能の制限により、変換されない寸法があります。

◦ ENF to JT

本機能を使用するには、JT ENF Reader 3Dアノテーション変換オプションが必要です。

項目	JT
注記	Note / Coordinate Note
デーラム	Datum Feature Symbol
デーラムターゲット	Datum Target
幾何公差	Feature Control Frame
仕上げ記号	Surface Finish
寸法	Dimension
溶接	Spot Weld / Line Weld
ロケータ	Locator
ユーザ定義シンボル	User Defined Symbol

◦ ENF to STEP

本機能を使用するには、STEP ENF Reader 3Dアノテーション変換オプションが必要です。

項目	STEP
注記	Note
デーラム	Datum
デーラムターゲット	Datum Target
幾何公差	Geometric Tolerance

項目	STEP
仕上げ記号	Surface Roughness
寸法	Dimension
溶接	Weld Symbol

- ENF to CADmeister (standalone)

項目	CADmeister
注記	3D注記
ポリライン寸法	直線要素としてポリラインの見た目に合わせて変換

- ENF to 3D PDF

項目	3D PDF
注記	Text
デーラム	Datum
デーラムターゲット	Datum
幾何公差	Gdt
寸法	Dimension
仕上げ記号	Roughness
ロケータ	Locator

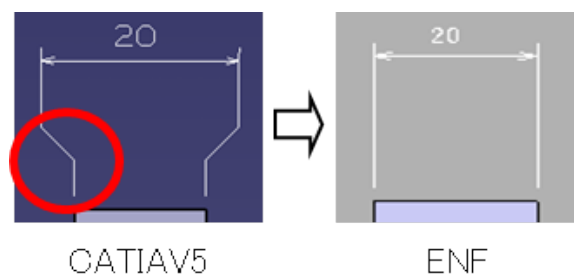
IV. 注意事項

1. CAD to ENF

CATIA V5 to ENF

CATIA V5の制限等により、正常に変換されない3Dアノテーションが存在します。

- 寸法線のファンネル表示



2. ENF to CAD

ENF to NX I-deas

NX I-deasの仕様の制限により、ENF to NX I-deasで変換されない3Dアノテーション要素が存在します。

下記の例の制限により、変換で3Dアノテーションの情報が変更、及び欠落した場合、ASFALISログに出力されます。

ログのフォーマット

```
PMI Warning: <PMIname>, <Contents>, <message>, <name> (x,y,z).
```

<PMIname>: PMI項目名

<Contents>: 変更、及び欠落した情報

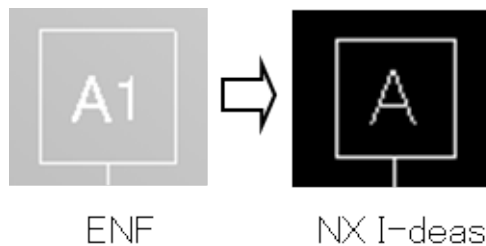
<Message>: 情報の変更、及び欠落内容

<Name>: PMIの名前

<x, y, z>: PMIの座標

◦ Datum Feature Symbol

ラベルのフォーマットは、アルファベット一文字であるため、合致しない情報は変換されません。

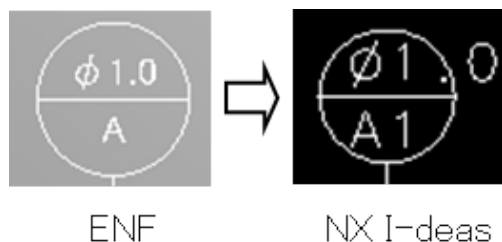


▪ ログ

```
PMI Warning: Datum, A1, Identifier was modified from (A1) to (A),
-----1 (13.59, 3.93, 36.95).
```

◦ Datum Target

ラベルのフォーマットはアルファベット一文字+通し番号 であるため、合致しない情報は変換されません。

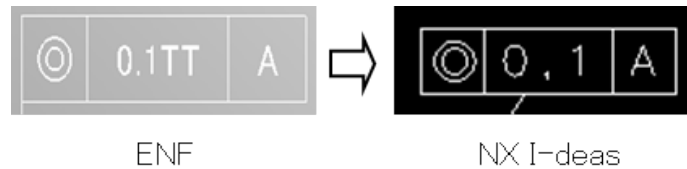


▪ ログ

PMI Warning: DatumTarget, |1.0 A, Identifier was modified from (A) to (A1),1 (7.78, -23.36, 40.45).

◦ Feature Control Frame

Feature Control Frameのフォーマットに合致しない情報は変換されません。

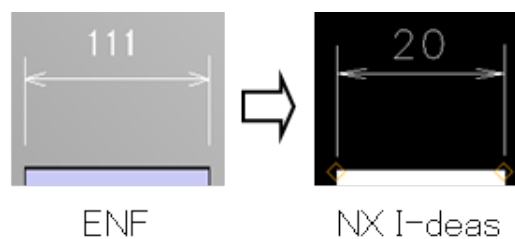


▪ ログ

PMI Warning: GD&T, 0.1 A, Value was modified from (TT) to (), Geometrical Tolerance.1 (0.98, 26.64, 36.95).

◦ Dimension

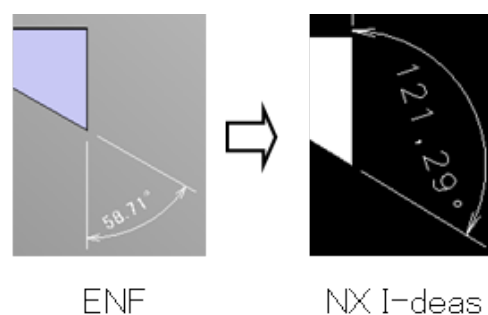
NX I-deasは、擬似寸法には対応していません。(NX I-deasは実寸値)



▪ ログ

PMI Warning: Dimension, 111, Dimension value was changed from (111) to (20), Dimension_1 (9.55, 29.52, -51.49).

NX I-deasのAPIの不具合により、角度寸法が正しく表示されません。
180 - Xの角度、または対角が表示されます。

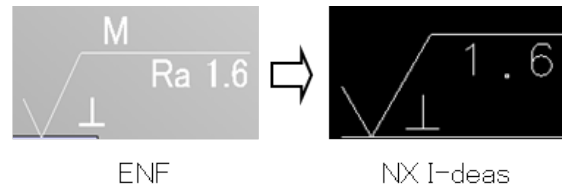


▪ ログ

PMI Warning: Dimension, 58.71`, Angular dimension value was changed to (180 - angle), Dimension_3 (-20.28, 20.00, -5.98).

◦ Surface Finish

フィールドは数値のみ対応。

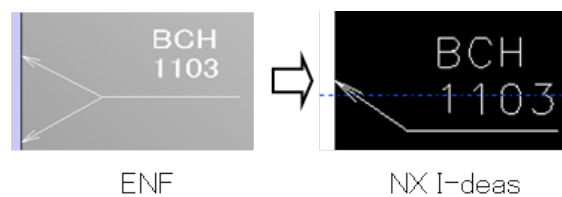


▪ ログ

PMI Warning: Roughness, M Ra 1.6, Value was modified from (M Ra 1.6) to (1.6), Roughness.1 (14.86, -11.21, -28.99).

◦ リーダーライン

NX I-deasは複数リーダーラインには対応していません。



▪ ログ

PMI Warning: Note, ??????????????, No Leader line was generated, _____.1 (206.00, 42.72, 35.87).

ENF to CADmeister (standalone)

注記について以下の制限があります。

- オンスクリーンは常にオン
- フレームは常になし
- 複数リーダーカーブには未対応。リーダーカーブの数だけ注記を作成
- 矢印タイプは常に"→"

V. キャプチャ・ビューの変換

1. CAD to ENF

◦ CATIA V5 to ENF

変換対象要素	詳細
Capture	<ul style="list-style-type: none"> * パート、アセンブリが管理しているCaptureを変換 * キャプチャが管理している、視点、アノテーション (FTA+溶接 記号) の表示/非表示属性、及び 1 平面タイプの断面(クリッピング平面)を変換
Named Views	<ul style="list-style-type: none"> * トップのアセンブリ・パートが管理しているNamed Viewsに登録されている視点方向を変換

◦ 3DEXPERIENCE to ENF

変換対象要素	詳細
Capture	<ul style="list-style-type: none"> * パートが管理しているCaptureを変換 * キャプチャが管理している、視点、アノテーションの表示/非表示属性および1平面タイプの断面(クリッピング平面)を変換
View (Camera)	<ul style="list-style-type: none"> * パートが管理しているViewsに登録されている視点方向を変換

◦ NX I-deas to ENF

変換対象要素	詳細
Model View	<ul style="list-style-type: none"> * パート、アセンブリが管理しているModel Viewを変換 * Model Viewが管理している断面は変換対象外 * アセンブリが管理しているModel Viewが保持しているPMIの表示/非表示属性は変換対象外

◦ NX to ENF

変換対象要素	詳細
Model View	<ul style="list-style-type: none"> * パート、アセンブリが管理しているModel Viewを変換 * Model Viewが管理している3D PMIの表示/非表示属性、レイヤの表示/非表示属性を変換 * Model Viewが管理している断面(PMI断面ビューの断面 およびビュー断面)を変換

◦ Creo Parametric to ENF

変換対象要素	詳細
Combined View	<ul style="list-style-type: none"> * パート、アセンブリが管理しているCombined Viewを変換 * Combined Viewが管理している、視点、断面及びアノテーションの表示/非表示属性を変換 * 断面についてはCreo Parametric 3.0以降、単一平面またはゾーンタイプの断面が対象 * 断面に作成されたハッチングは変換対象外
Orientation	<ul style="list-style-type: none"> * トップのアセンブリ・パートが管理しているOrientationに登録されている視点方向を変換

◦ SOLIDWORKS to ENF

変換対象要素	詳細
Annotation View	<ul style="list-style-type: none"> * トップのアセンブリ・パートが管理するAnnotation Viewを変換 * Annotation Viewが管理している視点、及びアノテーションの表示/非表示属性を変換
3D View	<ul style="list-style-type: none"> * トップのアセンブリ・パートが管理する3D Viewを変換 * 3D Viewが管理している視点、断面、及びアノテーションの表示/非表示属性を変換

◦ JT to ENF

変換対象要素	詳細
Model View	<ul style="list-style-type: none"> * パート、アセンブリが管理しているModel Viewを変換 * Model Viewが管理している視点、PMIの表示/非表示属性、要素(パート、アセンブリ、形状要素)の表示属性、及び断面を変換

◦ STEP to ENF

変換対象要素	詳細
Saved View	<ul style="list-style-type: none"> * パートが管理しているSaved Viewを変換 * Saved Viewが管理している視点、PMIの表示/非表示属性を変換 * パートが管理している1平面タイプの断面を変換

2. ENF to CAD

◦ ENF to CATIA V5

変換対象要素	詳細
Capture	<ul style="list-style-type: none"> * パートが管理しているCaptureを変換 * キャプチャが管理している、視点、及びアノテーション(FTA)の表示/非表示属性を変換 * パート、アセンブリが管理している断面を1平面タイプのクリッピング平面として変換

◦ ENF to 3DEXPERIENCE

変換対象要素	詳細
Capture	<ul style="list-style-type: none"> * パートが管理しているCaptureを変換 * キャプチャが管理している、視点、アノテーションの表示/非表示属性、および1平面タイプの断面(クリッピング平面)を変換

◦ ENF to NX

変換対象要素	詳細
Model View	<ul style="list-style-type: none"> * パート、アセンブリが管理しているカメラを変換 * カメラが保持する3D PMIの表示/非表示を変換 * アセンブリのカメラが保持する、配下パート中の3D PMIの表示/非表示は変換対象外 * パート、アセンブリが管理している断面をPMI断面ビューとして変換

◦ ENF to JT

変換対象要素	詳細
Model View	<ul style="list-style-type: none"> * パート、アセンブリが管理しているカメラを変換 * カメラが保持するPMIの表示/非表示属性を変換 * カメラが保持する要素(パート、アセンブリ、形状要素)の表示状態を変換
Section	* パート、アセンブリが持つ断面を変換

◦ ENF to STEP

変換対象要素	詳細
Saved View	<ul style="list-style-type: none"> * パートが管理しているカメラを変換 * パートが管理している1平面タイプの断面を変換

VI. 断面の変換

1. CAD to ENF

◦ NX I-deas to ENF

変換対象要素	詳細
Section	* Section名、切断平面の輪郭線、切断方向矢印をポリラインとして変換

VII. ハッチングの変換

1. CAD to ENF

◦ NX I-deas to ENF

変換対象要素	詳細
Hatching	<ul style="list-style-type: none"> * ハッチングをポリラインとして変換 * 断面に作成されたハッチングは変換対象外

◦ NX to ENF

変換対象要素	詳細
Hatching	<ul style="list-style-type: none"> * 平面に作成されたハッチングのみ変換対象 * ハッチングラインの線種はすべて実線として変換

3.4.2. 属性変換

I. ライセンス

属性をENF ReaderでCADに変換する際には、ENF Readerのライセンスの他に属性変換オプションのライセンスを必要とします。ライセンスコードについてはRequirementシート(Requirement.pdf)をご参照ください。(Writer側は標準機能)

II. 変換オプション

- 属性を変換するには、以下のオプションをパラメータファイルに記述します。

XConvertSystemProperty	システム属性を変換します。
XConvertUserProperty	ユーザ属性を変換します。
XConvertMaterial	材料属性を変換します。

III. 変換対象

属性マッピング表 / 変換対象要素

1. ENF Writer

- CATIA V5 to ENF

CATIA V5属性名	ENF キー名称	S/U
名称	PartName	S
パーツ番号	PartNumber	S
改訂	PartRevision	S
定義	PartDefinition	S
記述	Description	S
ソース	PartSource	S
パートのレイヤ	PartLayer	S
インスタンスの記述	InstanceDescription	S
マテリアル名	Material	S
ユーザ属性	(ユーザ定義)	U

- 属性変換対象要素：CATProduct、CATPart、Shell(Materialのみ)、Lump(Materialのみ)、形状セット(ユーザ属性のみ)、ユーザ定義フィーチャ(ユーザ属性のみ)

- 3DEXPERIENCE to ENF

3DEXPERIENCE 属性名	ENF キー名称	S/U
名称	PartName	S
説明	Description	S
リビジョン	PartRevision	S
最終更新日	NativeFileLastWriteTime	S
ユーザ属性	(ユーザ定義)	U

- 属性変換対象要素：Product、3D Part

◦ NX I-deas to ENF

NX I-deas属性名	ENF キー名称	S/U
パート名/アセンブリ名	PartName	S
パーツ番号	PartNumber	S
Configの名前	ConfigName	S
Bin名	BinName	S
バージョン	PartVersion	S
リビジョン	PartRevision	S
説明	Description	S
履歴	ChangeHistory	S
マテリアル名	Material	S
ユーザ属性	(ユーザ定義)	U

- 属性変換対象要素：Assembly, Part

◦ NX to ENF

NX属性名	ENF キー名称	S/U
マテリアル名	Material	S
変換元ファイル最終更新時刻	NativeFileLastWriteTime	S
変換元ファイルサイズ	NativeFileSize	S
ユーザ属性	(ユーザ定義)	U

- 属性変換対象要素：Part、Body(SolidBody、SheetBody)(*）、Face、Edge、Curve、Point
(*): MaterialはBodyのみ

◦ Creo Parametric to ENF

Creo Parametric属性名	ENF キー名称	S/U
リビジョン	PartRevision	S
マテリアル	Material	S
パラメータ	(ユーザ定義)	U

- 属性変換対象要素：Assembly、Part、Datum Point、Edge、Surface、Annotation Feature

◦ Inventor to ENF

Inventor属性名	ENF キー名称	S/U
タイトル	Title	U
表題	Subject	U
作成者	Author	U
管理者	Manager	U
会社	Company	U
カテゴリ	Category	U
キーワード	Keywords	U
コメント	Description	S
ファイルサブタイプ	Document SubType Name	U
ストック番号	Stock Number	U
説明	PartComment	S
履歴番号	PartRevision	S
プロジェクト	Project	U
設計者	Designer	U
エンジニア	Engineer	U
責任者	Authority	U
コストセンター	Cost Center	U
見積もり費用	Cost	U
ステータス	User Status	U
設計ステータス	Design Status	U
確認者	Checked By	U
確認日	Date Checked	U
エンジニアリング承認者	Engr Approved By	U

Inventor属性名	ENF キー名称	S/U
エンジニアリング承認日	Engr Date Approved	U
製造承認者	Mfg Approved By	U
製造承認日	Mfg Date Approved	U
Webリンク	Catalog Web Link	U
ベンダー	Vendor	U
材料名	Material	S
ユーザ属性	(ユーザ定義)	U

- 属性変換対象要素：Assembly、Part(MaterialはPartのみ)

◦ SOLIDWORKS to ENF

SOLIDWORKS属性名	ENF キー名称	S/U
ユーザー定義プロパティ	(ユーザ定義)	U

- 属性変換対象要素：Assembly、Part

◦ JT to ENF

JT属性名	ENF キー名称	S/U
ユーザ属性	(ユーザ定義)	U

- 属性変換対象要素：Assembly、Part、BodyNode

◦ STEP to ENF

STEP属性名	ENF キー名称	S/U
id (PRODUCT)	PartNumber	S
name (PRODUCT)	PartName	S
description (PRODUCT)	Description	S
id (PRODUCT_DEFINITION_FORMATION)	PartVersion	S
description (PRODUCT_DEFINITION_FORMATION)	PartComment	S
id (PRODUCT_DEFINITION)	PartDefinition	S
make_or_buy (PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE)	PartSource	S
変換元ファイル最終更新時刻	NativeFileLastWriteTime	S
変換元ファイルサイズ	NativeFileSize	S

STEP属性名	ENF キー名称	S/U
ユーザ属性	(ユーザ定義)	U

- 属性変換対象要素：Assembly、Part
- STEP AP242 BOM to ENF

STEP AP242 BOM属性名	ENF キー名称	S/U
Part/Name/CharacterString (*1)	PartName	S
Part/Identifier.id (*1)	PartName もしくは PartNumber	S
PartVersion/Identifier.id (*2)	PartRevision	S
変換元ファイルのパス	NFName	S
PropertyValueAssignment	(ユーザ定義)	U (*3)

(*1) PartIDMappingパラメータのルールに従ってセットされます。

(*2) PartVersionIDMappingパラメータのルールに従ってセットされます。

(*3) 外部参照のパートとXML側で重複した属性がある場合は、どちらを優先するかをXConvertUserPropertyのパラメータで指定することが可能です。

- 属性変換対象要素：Assembly、Part
- CADmeister to ENF (stand alone)

CADmeister属性名	ENF キー名称	S/U
部品属性	(ユーザ定義)	U

- 属性変換対象要素：オブジェクト、複合図形

2. ENF Reader

- ENF to CATIA V5

ENF キー名称	S/U	CATIA V5 属性名
PartName	S	名称
PartRevision	S	改訂
PartDefinition	S	定義
Description	S	記述
PartSource	S	ソース
PartLayer	S	パートのレイヤ
InstanceDescription	S	インスタンスの記述
Material	S	マテリアル名

ENF キー名称	S/U	CATIA V5 属性名
(ユーザ定義)	U	ユーザ属性

- 属性変換対象要素：CATProduct、CATPart、Shell (Materialのみ)、Lump (Materialのみ)

◦ ENF to NX I-deas

ENF キー名称	S/U	NX I-deas 属性名
PartNumber	S	パーツ番号
PartRevision	S	リビジョン
BinName	S	Bin名
Description	S	説明
ChangeHistory	S	履歴
Material	S	マテリアル名
(ユーザ定義)	U	ユーザ属性

- 属性変換対象要素：Assembly、Part

◦ ENF to NX

ENF キー名称	S/U	NX 属性名
PartNumber	S	ユーザ属性Title "PART_NUMBER"
PartName	S	ユーザ属性Title "PART_NAME"
PartRevision	S	ユーザ属性Title "PART_REVISION"
PartDefinition	S	ユーザ属性Title "PART_DEFINITION"
Description	S	ユーザ属性Title "PART_DESCRIPTION"
PartSource	S	ユーザ属性Title "PART_SOURCE"
ChangeHistory	S	ユーザ属性Title "PART_CHANGE_HISTORY"
PartComment	S	ユーザ属性Title "PART_COMMENT_NOTE"
ConfigName	S	ユーザ属性Title "PART_CONFIGURATION"
PartVersion	S	ユーザ属性Title "PART_VERSION"
Material	S	ユーザ属性Title "PART_MATERIAL"
Material	S	マテリアル名
(ユーザ定義)	U	ユーザ属性

- 属性変換対象要素：Assembly、Body (SolidBody、SheetBody)(*）、Face、Curve、Point、データム平面

(*): MaterialはBodyのみ

◦ ENF to Creo Parametric

ENF キー名称	S/U	Creo Parametric 属性名
PartNumber	S	名前が"PARTNUMBER"のパラメータ
PartName	S	名前が"PARTNAME"のパラメータ
PartRevision	S	名前が"PARTREVISION"のパラメータ
PartDefinition	S	名前が"PARTDEFINITION"のパラメータ
Description	S	名前が"DESCRIPTION"のパラメータ
PartSource	S	名前が"PARTSOURCE"のパラメータ
Material	S	名前が"MATERIAL"のパラメータ
ConfigName	S	名前が"CONFIGNAME"のパラメータ
NFName	S	名前が"NATIVEFILENAME"のパラメータ
PartVersion	S	名前が"PARTVERSION"のパラメータ
TargetFileName	S	名前が"TARGETFILENAME"のパラメータ
PartLayer	S	名前が"PARTLAYER"のパラメータ
ChangeHistory	S	名前が"CHANGEHISTORY"のパラメータ
(ユーザ定義)	U	パラメータ

- 属性変換対象要素：Assembly、Part

◦ ENF to Inventor

ENFキー名称	S/U	Inventor属性名
Title	U	タイトル
Subject	U	表題
Author	U	作成者
Manager	U	管理者
Company	U	会社
Category	U	カテゴリ
Keywords	U	キーワード
Description	S	コメント
Stock Number	U	ストック番号
PartComment	S	説明
PartRevision	S	履歴番号

ENFキー名称	S/U	Inventor属性名
Project	U	プロジェクト
Designer	U	設計者
Engineer	U	エンジニア
Authority	U	責任者
Cost Center	U	コストセンター
Cost	U	見積もり費用
User Status	U	ステータス
Checked By	U	確認者
Date Checked	U	確認日
Engr Approved By	U	エンジニアリング承認者
Engr Date Approved	U	エンジニアリング承認日
Mfg Approved By	U	製造承認者
Mfg Date Approved	U	製造承認日
Catalog Web Link	U	WEBリンク
Vendor	U	ベンダー
Material	S	材料名
(ユーザ定義)	U	ユーザ属性

- 属性変換対象要素：Assembly、Part(MaterialはPartのみ)

◦ ENF to SOLIDWORKS

ENFキー名称	S/U	SOLIDWORKS属性名
(ユーザ定義)	U	ユーザ定義プロパティ

- 属性変換対象要素：Assembly、Part、Body (*）、Face (*）、Edge (*)
(*): マクロ・APIでアクセスできるIAttributeという属性に変換される

◦ ENF to JT

ENFキー名称	S/U	JT属性名
PartName	S	\$PartName
PartNumber	S	\$PartNumber(変換元がCATIA V5以外) \$Nomenclature(変換元がCATIA V5の場合)
NFName	S	\$NativeFileName
ConfigName	S	\$ConfigName

ENFキー名称	S/U	JT属性名
BinName	S	\$BinName
ChangeHistory	S	\$ChangeHistory
Description	S	\$Description
PartComment	S	\$PartComment
PartDefinition	S	\$PartDefinition
PartVersion	S	\$PartVersion
PartRevision	S	\$PartRevision
PartSource	S	\$PartSource
PartLayer	S	\$PartLayer
NativeFileLastWriteTime	S	\$NativeFileLastWriteTime
NativeFileSize	S	\$NativeFileSize
InstanceDescription	S	\$InstanceDescription
Material	S	\$Material
(ユーザ定義)	U	ユーザ属性

- 属性変換対象要素：Assembly、Part、BodyNode(V5の形状セット、ユーザ定義フィーチャの属性をBodyNodeのパラメータに変換する)

◦ ENF to STEP

ENF キー名称	S/U	STEP属性名
PartNumber	S	id (PRODUCT)
PartName	S	name (PRODUCT)
Description	S	description (PRODUCT)
PartVersion	S	id (PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE)
PartComment	S	description (PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE)
PartDefinition	S	id (PRODUCT_DEFINITION)
PartSource	S	make_or_buy (PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE)
(ユーザ定義)	U	ユーザ属性

- 属性変換対象要素：Assembly、Part

◦ ENF to STEP AP242 BOM

ENF キー名称	S/U	ENF to STEP AP242 BOM属性名
(ユーザ定義)	U	PropertyValueAssignment

- 属性変換対象要素：Assembly、Part
- ENF to CADmeister (standalone)

ENF キー名称	S/U	CADmeister属性名
(ユーザ定義)	U	部品属性

- 属性変換対象要素：オブジェクト、複合図形



1. S/U System属性/User属性
2. System属性は、CADやシステムによって定義されているプロパティ
キーの編集が不可能なプロパティ
3. User属性は、Key=Value の形で自由に定義可能なプロパティ
4. CATIA V5 to/from NX I-deasの変換では、V5のパーツ番号とNX I-deasのパーツ
番号、V5の名称とNX I-deasのパート名が対応するよう、ヒーリングで属性の
付け替えを行う

IV. 属性のマッピング

- ENFに定義されている属性名、属性の値などの変更(マッピング)が可能です。
「[3.3.4, “Attribute Editor”](#)」の説明をご確認ください。

3.4.3. ライセンス管理

ASFALIS製品はエリジオンのライセンス管理プログラムによりライセンス管理されます。
新しいライセンス管理プログラムの使用方法等については、別冊の「Sentinel RMS License Manager
セットアップ&クイックスタートガイド」をご参照下さい。

I. 製品コード

各コンポーネントはエリジオンより提供されるASFALIS専用のライセンスを参照して稼動します。各
コンポーネントのライセンスフィーチャは別冊の「Requirement.pdf」をご参照ください。

II. ライセンスの取得/解放

ここではENF Writer/Readerを実行した際に起こるライセンスの取得と解放の流れを説明します。

(1) ENF Writerのみ(CAD A to ENF)を実行する場合

CAD AのCAD IDを1XXとするとelybatchはプロダクトID 1XX001として実行されます。

1. elybatchを-p1XX001またはシナリオ<productcode id="1XX001"/>で実行
2. elybatchはelybatchのライセンス"ASF-ELYBATCH"の存在のチェックとENF Writerのライセン

ス"ASF-<CAD A>-ENFS"を取得

3. elybatchは取得したENF Writerのライセンス"ASF-<CAD A>-ENFS"をENF Writer本体に継承
4. ENF Writerはelybatchから継承されたライセンスを使用して処理を実行
5. ENF Writerは処理が終わったらelybatchに終了を通知
6. elybatchはENF Writerのライセンス"ASF-<CAD A>-ENFS"を解放

(2) ENF Writer、Healing、ENF Readerを連続実行(CAD A to CAD B)する場合

CAD AのCAD IDを1XX、CAD BのCAD IDを1YYとするとelybatchはプロダクトID 1XXYY1として実行されます。

1. elybatchを-p1XXYY1またはシナリオ<productcode id="1XXYY1"/>で実行
2. elybatchはelybatchのライセンス"ASF-ELYBATCH"の存在のチェックとENF Writerのライセンス"ASF-<CAD A>-ENFS"取得
3. ENF Writerはelybatchの子プロセスとしてライセンス"ASF-<CAD A>-ENFS"を使用し処理を実行
4. ENF Writerは処理が終わったらelybatchに終了を通知
5. elybatchはENF Writerのライセンス"ASF-<CAD A>-ENFS"を解放
6. elybatchはENF Readerのライセンス"ASF-ENFS-<CAD B>"を取得
7. Healerはelybatchの子プロセスとしてライセンス"ASF-ENFS-<CAD B>"を使用し処理を実行
8. Healerは処理が終わったらelybatchに終了を通知
9. elybatchはENF Readerのライセンス"ASF-ENFS-<CAD B>"を解放
10. elybatchはENF Readerのライセンス"ASF-ENFS-<CAD B>"を取得
11. ENF Readerはelybatchの子プロセスとしてライセンス"ASF-ENFS-<CAD B>"を使用し 処理を実行
12. ENF Readerは処理が終わったらelybatchに終了を通知
13. elybatchはENF Readerのライセンス"ASF-ENFS-<CAD B>"を解放

(3) ENF Reader(ENF to CAD B)でオプションライセンスが必要な処理を実行する場合

CAD BのCAD IDを1YYとするとelybatchはプロダクトID 100YY1として実行されます。

ENF Readerのライセンス"ASF-ENF-<CAD B>"に加え、3Dアノテーション変換を実行する場合 "ASF-ANN/<CAD B>"、属性変換を実行する場合 "ASF-ATR/<CAD B>"のライセンスも同時に使用されます。

以下は属性変換を行う場合の例です。elybatchがオプションライセンスを取得できない場合は変換エラーとなります。

1. elybatchを-p100YY1またはシナリオ<productcode id="100YY1"/>で実行
2. elybatchはelybatchのライセンス"ASF-ELYBATCH"の存在のチェックとENF Readerのライセンス"ASF-ENFS-<CAD B>"および属性変換オプションのライセンス"ASF-ATR/<CAD B>"を取得
3. elybatchは取得したENF Readerのライセンス"ASF-ENFS-<CAD B>"と属性変換オプションのラ

イセンス"ASF-ATR/<CAD B>"をENF Reader本体に継承

4. ENF Readerはelybatchの子プロセスとしてライセンス"ASF-ENFS-<CADB>"と"ASF-ATR/<CAD B>"を使用し処理を実行
5. ENF Readerは処理が終わったらelybatchに終了を通知
6. elybatchはENF Readerのライセンス"ASF-ENFS-<CADB>"と属性変換オプションのライセンス"ASF-ATR/<CAD B>"を解放

4. 変更履歴

2019年11月29日 ASFALIS EX8.2 beta版

- 2.2.1 コンフィグファイルキーワード
 - 3DEXPERIENCE 関連の項目をプロダクトセクションに追加
- 3.1 Adapter
 - CATIA V5(CAA)、NX(UFUNC)、Creo Parametric で新アダプタを正式サポート
 - 3DEXPERIENCE、3DXML を追加
- 3.1.2. 3DEXPERIENCE to ENF / ENF to 3DEXPERIENCE
 - 新規追加
- 3.1.25. 3DXML to ENF
 - 新規追加
- 3.4.1. 3Dアノテーション変換
 - 3DEXPERIENCE to ENF/ENF to 3DEXPERIENCE を新規に追加
- 3.4.2. 属性変換
 - 3DEXPERIENCE to ENF を新規に追加

2018年12月21日 ASFALIS EX8.1 beta版

- 2.2.1 コンフィグファイルキーワード
 - CATIAV5_EnvPath を追加
- 3.1.1 CATIA V5 to ENF / ENF to CATIA V5
 - 注意点を更新
- 3.1.4 Creo Parametric to ENF / ENF to Creo Parametric
 - 注意点を更新
 - 制限事項を更新
- 3.3.1 PDQChecker
 - チェック結果PDF レポートの説明を追加
- 3.4.2 属性変換
 - STEP AP242 BOM to ENF/ENF to STEP AP242 BOM の対応表を追加。

2018年9月12日 ASFALIS EX8.0.14 正式版

- 3.1.4 Creo Parametric to ENF / ENF to Creo Parametric
 - 制限事項を更新

2018年4月6日 ASFALIS EX8.0.14 正式版

- 3.3.3 CAD Validator

- 各種レポートファイルの差異をまとめた表を追加

2018年1月26日 ASFALIS EX8.0 beta版

- 2.2.1 コンフィグファイルキーワード
 - NUM_OF_PROCESSES の説明を更新
- Geometry Validator 関連の表記を変更
 - Geometry Validator → CAD Validator
 - 形状比較 → CAD 比較
- 3.1 Adapter
 - Creo Parametric について更新
 - JT について更新
- 3.1.5 NX to ENF / ENF to NX
 - 制限事項を更新
- 3.1.14 JT to ENF / ENF to JT
 - 制限事項を更新
- 3.1.17 3.1.17 CADmeister to ENF / ENF to CADmeister (standalone)
 - 注意点を更新
- 3.1.23 ENF to 3D PDF
 - 制限事項を更新
- 3.3.2 Geometry Simplifier
 - 概要説明を更新
 - 「見えないボリユームの除去機能」について説明および模式図を更新
- 新規対応コンポーネントの説明を追加
 - 3D PDF Editor

2017年5月19日 ASFALIS EX7.2 beta版

- 新規対応コンポーネントの説明を追加
 - ENF to 3D PDF
- 3.1.13 STEP to ENF / ENF to STEP
 - 注意点追記
- 3.4.1 3Dアノテーション変換
 - 変換対象3DアノテーションにSTEP to ENF/ENF to STEPの対応表を追記
 - キャプチャー・ビューの変換にSTEP to ENF/ENF to STEPの対応表を追記
- 3.4.2 属性変換
 - STEP to ENF/ENF to STEPの対応表を追記

2016年8月19日 ASFALIS EX7.1 beta版

- 2.2.1 コンフィグファイルキーワード
 - 共通セクションにもBATCH_TIME_OUTを追加
- 2.4.1 シナリオ書式
 - GEOMDIFF 注意点 更新/追記
- 3.1.7 SOLIDWORKS to ENF / ENF to SOLIDWORKS
 - 注意点2、4を更新
- 3.1.8 Inventor to ENF / ENF to Inventor
 - 注意点2を更新
- 3.3.3 CAD Validator
 - IV. 制限事項を追加
 - V. カスタマイズファイルの設定を追加
 - 比較レポートファイルの動作・制限等に追記
- 3.4.1 3Dアノテーション変換
 - 変換オプションを更新
 - NXについて更新
 - JTについて更新

2016年1月8日 ASFALIS EX7.0 正式版

- 2.2.1 コンフィグファイルキーワード
 - PRO_PROD_DCU_FINISHを追加
- 3.3.2 Geometry Simplifier
 - 微小ボリユームの除去機能を追加
- 3.4.1 3Dアノテーション変換
 - 変換対象3DアノテーションにCreo Parametric to ENFの対応表を追記

2015年11月4日 ASFALIS EX7.0 beta版

- CADの表記を変更
 - Pro/ENGINEER → Creo Parametric
 - CoCreate → Creo Elements/Direct
 - SolidWorks → SOLIDWORKS
 - I-deas → NX I-deas
- アダプタ名称内のSAを(standalone)に変更
- 新規対応コンポーネントの説明を追加
 - STEP AP242 BOM ENF Writer / ENF Reader

- ENF Reader(新アダプタ)共通の仕様変更の更新
 - CADのファイル名として不正な名前(字数制限超え、使用禁止文字)が指定された場合、出力ファイル名を変更して正常終了するようになりました。
- 2.2.1 コンフィグファイルキーワード
 - NUM_OF_PROCESSESを修正
- 2.4 XML シナリオ
 - セクション: GEOMDIFFを修正
- 3.1.1CATIA V5 to ENF / ENF to CATIA V5
 - 必要実行モジュール等の修正
 - 注意点1, 4を修正
 - 制限事項1を修正
- 3.1.5NX to ENF / ENF to NX
 - 設定方法、注意点、制限事項を修正
 - 線幅マッピングファイルについて修正
- 3.1.7 SolidWorks to ENF / ENF to SolidWorks
 - 注意点4を更新
- 3.1.5 iCAD to ENF / ENF to iCAD
 - 制限事項9、10を更新
- 3.1.18 CADmeister to ENF / ENF to CADmeister (standalone)
 - 注意点1を修正
- 3.3.2 Geometry Simplifier
 - 曲線群の平滑化を追加
- 3.3.3 CAD Validator
 - XMLログについて更新
 - V . 比較レポートファイル (3D HTML形式)を追加
 - VI . 比較レポートファイル (3D PDF形式)
- 3.4.1 3D アノテーション変換
 - NXについて更新
 - CATIA V5について更新
 - JTについて更新

2014年11月11日 ASFALIS EX6.1 beta版

- 新規対応コンポーネントの説明を追加
 - PLM XML ENF Writer / ENF Reader

- iCAD ENF Writer / ENF Reader
- 3.1.5 NX to ENF / ENF to NX
 - 注意点1を修正
 - 制限事項2を修正
- 3.1.17 CADmeister to ENF / ENF to CADmeister (standalone)
 - 必要実行モジュール等の修正
 - 注意点1, 5を修正

2014年5月13日 ASFALIS EX6.0 正式版

- 2.2 コンフィグファイル
 - 共通セクションとプロダクトセクションの説明を追記
- 2.2.1 コンフィグファイルキーワード
 - LEGACY_MODEの記載箇所を変更
変更前) 共通セクション
変更後) プロダクトセクション(プロダクト共通)
- 2.3.1 引数
 - 引数-uの説明追記
- 3.1 Adapter
 - アダプタの呼称変更
変更前) アダプタ / 旧アダプタ
変更後) 新アダプタ / 旧アダプタ
- デフォルトで使用するアダプタを変更
CATIA V5(CAA), NX, JTのENF Readerは新アダプタをデフォルトに、
他は旧アダプタをデフォルトにしました。
上記以外の新アダプタは「使用可能だがサポート対象外」の扱いになりました。
- 3.1.5 NX to ENF / ENF to NX
 - 注意点3,4,5を追記
- 3.1.17 CADmeister to ENF / ENF to CADmeister (standalone)
 - 制限事項2を修正、3を追記

2014年3月7日 ASFALIS EX6.0 beta版

- マニュアルの構成を一部変更
- 新規対応コンポーネントの説明を追加
 - CADmeister SA ENF Writer / ENF Reader
 - CATIA V5 SA ENF Writer / ENF Reader
 - Pro/ENGINEER SA ENF Writer
 - NX SA ENF Writer

- ENF Editor
- 以下のアダプタについて、アダプタと旧アダプタの記述を追加
 - CATIA V5(CAA) ENF Writer / ENF Reader
 - NX ENF Writer / ENF Reader
 - Pro/ENGINEER Creo Parametric ENF Writer / ENF Reader
 - Autodesk Inventor ENF Writer / ENF Reader
 - SolidWorks ENF Writer / ENF Reader
 - JT ENF Writer / ENF Reader
- 以下のコンフィグキーワードを追加
 - LEGACY_MODE
 - NUM_OF_PROCESSES
- シナリオタグの変更
 - セクション： GEOMDIFF
<enfdifffile>タグを廃止、<diffresultfile>を追加しました。
- 3.1.1 CATIA V5 to ENF / ENF to CATIA V5
 - 注意点1 記述変更
変更前) LUM(License Use Runtime)
変更後) LUM(License Use Runtime)またはDSL(SDSS(Dassault Systemes License Server)
 - 注意点4 記述変更
- 3.1.4 Creo Parametric to ENF / ENF to
 - 注意点3 記述変更
変更前) ENF to Pro/E で、Pro/Eのコンフィグ設定(config.sup または config.pro)に
変更後) Pro/Eのコンフィグ設定(config.sup または config.pro)に
ENF Writer側でも変換に失敗する事例が確認されたため。
- 3.1.5 NX to ENF / ENF to NX
 - 注意点1 記述変更
変更前) UGS_LICENSE_SERVER
変更後) NX7.5 – NX8.5: UGS_LICENSE_SERVER / NX9.0: SPLM_LICENSE_SERVER
- 3.1.7 SolidWorks to ENF / ENF to SolidWorks
 - SolidWorksアダプタ(旧アダプタ)の配置されているフォルダがCADバージョン名のフォルダに変更になりました。コンフィグキーワードFROM_TRANSLATOR / TO_TRANSLATORで指定する、「アダプタ本体のあるディレクトリ」を変更してください。
変更前) C:\elysium\bin\SwExe\win
変更後) C:\elysium\bin\SwExe\win\2013
- 注意点4 記述変更
変更前) CAD側の問題は、SolidWorks 2013 SP4 で修正される予定です。
変更後) CAD側の問題は、SolidWorks 2013 SP4 で修正されています。

- 3.1.9 Creo Elements/Direct to ENF / ENF to
 - 入出力仕様変更
入力ファイルとして*.sda, *.sdp, *.sdファイルに対応しました。
- 3.1.12 STEP to ENF / ENF to STEP
 - 「ENF to STEPのプロトコルおよびファイルバージョンに関する注意点」を注意点欄に移動
 - 制限事項3を追記
- 3.1.14 JT to ENF / ENF to JT
 - 「ENF to JTのファイルバージョンに関する注意点」を注意点欄に移動
- 3.3.3 CAD Validator
 - 高速表示用比較に対応し関連する記述を追加
 - XMLログの出力内容を追加
新パラメータ、高速表示用比較
 - <enfdifffile>タグを廃止、<diffresultfile>タグを追加
 - 形状比較レポートファイルで出力されるxslファイルの名前を変更
 - レポート表示でサポートするInternet Explorerバージョンを変更
変更前) Internet Explorer 6 - 9
変更後) Internet Explorer 8 - 11
- JT to ENFで3Dアノテーション変換に対応し、関連する記述を追加
- 3.4.1 3Dアノテーション変換
 - 変換対象3DアノテーションにJTtoENFの対応表を追記
 - 変換対象3Dアノテーションの対応表でNXのPMI表記変更
NX to ENFのDatum Targetは新規対応
NX to ENFのFeature Control Frameは変換先をNoteからGDTへ変更
 - 変換対象3Dアノテーション キャプチャ・ビューの変換でCATIA V5 to ENFの記述を変更
削除) トップのアセンブリ・パートが管理しているCaptureのみ変換
削除) 視点が設定されていないキャプチャは変換対象外
誤記修正
- 3.4.2 属性変換
 - 以下の属性に対応
CATIA V5 to ENF インスタンスの記述
ENF to CATIA V5 インスタンスの記述
ENF to JT ENFのInstanceDescription、Material
 - ENF to STEPの属性マッピング表を追記
- 3.4.3 ライセンス管理
 - ライセンスコード表を削除
別冊の「Requirement.pdf」をご参照ください。
 - ENF Readerでオプションライセンスが必要な処理を実行する場合の例を追記

- ENF Readerで3Dアノテーション変換/属性変換を行う際にそれらのオプションライセンスが取得できない場合の動作を変更しました。

変更前) ENF Readerは実行されるが、3Dアノテーション/属性は変換されない

変更後) ライセンスエラーとなり変換が実行されない

本コンテンツに関わる著作権は株式会社エリジオンもしくは原権利者に帰属しています。
著作権者の承諾なしに無断で改変、複製、転載、再配布、転送、公衆送信、販売、貸与などの
行為をすることは禁じられています。