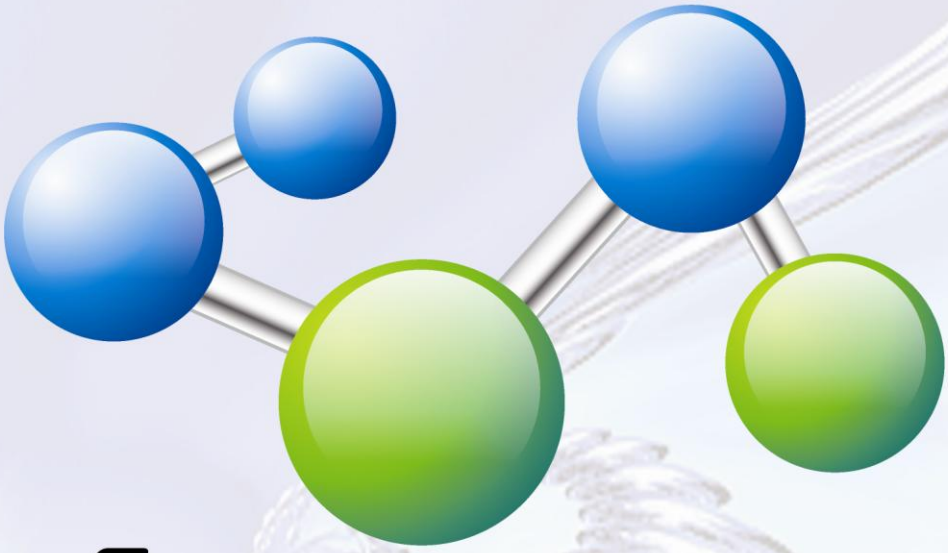

asteria warp

Business Automation Platform

環境移行ガイド

infoteria

The Information Cafeteria

ご注意

2013 年 1 月 21 日

インフォテリア株式会社

目次

1. 移行する前に【必ずお読みください】

ASTERIA 3 のプロジェクトファイルは、ASTERIA WARP Flow Designer で作成するプロジェクトファイルと互換性がありません。

ASTERIA 3 のプロジェクトファイル、データ、ASTERIA システムに関連するファイルを ASTERIA WARP Flow Service に移行し、動作のみ行う場合には ASTERIA 3 のプロジェクトファイルのままで運用できます（①）。ただし、ASTERIA WARP Flow Service に移行したあと、プロジェクトファイルを編集する（②）ためには ASTERIA 3 のプロジェクトファイルを ASTERIA WARP Flow Service のプロジェクトファイルに変換する必要があり、変換のためのツールが ASTERIA WARP インストーラーの中に含まれています。

上記①のように、そのまま運用する場合の移行手順は、後述の「2.フロー実行環境の移行手順」で説明します。上記②のように、プロジェクトファイルを変換してから運用するには、①の移行手順の途中から後述の「3.ASTERIA WARP に対応したプロジェクトファイルの変換する手順」で説明する手順を行ってください。

補足：
ASTERIA 3 のプロジェクトファイルの編集は ASTERIA Designer 3 でのみ可能です。そのまま運用する場合でも、ASTERIA WARP Flow Service に ASTERIA Designer 3 を接続して編集することはできません。また、ASTERIA Server 3 に ASTERIA WARP Flow Designer を接続して編集することはできません。プロジェクトファイルを編集するときには、サーバー、デザイナーのバージョンを合わせるようにしてください。

これ以降の製品名称は以下のように記述します。
- ASTERIA Server 3　→　**ASTERIA3 サーバー**
- ASTERIA Designer 3　→　**ASTERIA3 デザイナー**
- ASTERIA WARP Flow Service　→　**フローサービス**
- ASTERIA WARP Flow Designer　→　**フローデザイナー**

また、それぞれのインストールフォルダを以下のように記述します。
- ASTERIA Server 3 のインストールフォルダ　→　**ASTERIA3_HOME**
- ASTERIA WARP のインストールフォルダ　→　**WARP_HOME**
- ASTERIA WARP のデータフォルダ　→　**DATA_HOME**

（データフォルダとはインストール時に指定するデータ格納用フォルダで、初期値はインストールするドライブ直下の asteriahome となります。例：C:/asteirahome）

1.1. ASTERIA WARP のインストールについて

ASTERIA WARP は既に ASTERIA 3 がインストールされているマシンに対してもインストールすることができます。同一マシンで ASTERIA 3 と ASTERIA WARP を同時に起動することがある場合、インストール時に ASTERIA WARP で使用するポート番号は ASTERIA 3 とは異なるポート番号を指定してください。同一マシンで ASTERIA 3 と ASTERIA WARP を同時に起動することがない場合、同じポート番号を使用することができます。

2. フロー実行環境の移行手順

2.1. 移行手順

移行する手順は以下のとおりです。次項から詳細な手順を説明します。

1）ASTERIA 3 から実行設定をエクスポートする
2）ASTERIA WARP でアカウントを作成する
3）システムに関連するファイルをコピーする
4）フローに関連するデータ類をコピーする
※プロジェクトファイルの変換を行う場合、ここで変換作業を行います。
5）ASTERIA WARP で実行設定をインポートする
6）管理コンソールで必要事項を設定する
7）ASTERIA WARP を再起動する

注意：
ワークグループ機能を使って作成したプロジェクトファイルの場合、後述する「2.2.ワークグループからの移行」を参照してください。

1）ASTERIA 3 から実行設定をエクスポートする

1. ASTERIA3 サーバー、ASTERIA3 デザイナーを起動します。
2. ASTERIA3 デザイナーから移行対象のユーザーで接続します。
3. 実行設定の画面から「エクスポート」をクリックします。
4. エクスポートの画面では「全プロジェクトを出力」をチェックし、エンコーディングは shift_jis を指定します。

すべてのプロジェクト、実行設定を移行する必要がない場合は生成された fcs ファイルから対象外のプロジェクトの部分を削除してください。

この作業が終了した後は ASTERIA3 サーバー、ASTERIA3 デザイナーを停止できます。

2）ASTERIA WARP でアカウントを作成する

1. フローサービスを起動します。
2. フローサービスの管理コンソールから移行対象のユーザーを新規に作成します。

作成するユーザー名は必ずしも ASTERIA 3 でのドメイン、ユーザー名と同じでなくても構いませんが同じにすることを推奨します。ASTERIA 3 でのユーザーのホームディレクトリを ASTERIA3_HOME/home 以外に指定していた場合、ASTERIA WARP で作成するユーザーのホームディレクトリは必ず ASTERIA 3 で使用していたディレクトリとは別のディレクトリを指定してください。（デフォルトで作成した場合は別になります。）

ASTERIA 3 で URL 実行設定を使用している場合、必ずユーザーのコンテキスト設定を ASTERIA 3 での設定と同じにしてください。

3）システムに関連するファイルをコピーする

システムに関連する各種設定ファイルや JDBC ドライバ類を ASTERIA3_HOME 内から WARP_HOME 内へコピーします。
コピー対象のファイルとコピー先は以下のとおりです。

・ システムコネクション
ASTERIA3_HOME/conf/connections/asconnections.xconf
→ WARP_HOME/flow/conf/connections/asconnections.xconf
・ JDBC ドライバ
ASTERIA3_HOME/jre/lib/ext にある JDBC ドライバの jar（または zip）ファイル
→ WARP_HOME/jre/lib/ext

・ DTD
ASTERIA 3_HOME/services/flowservice/dtd フォルダ
→ WARP_HOME/flow/services/flowservice/dtd
（フローの中で DTD を使用していない場合、フォルダ自体が存在しないので移行は不要です。）

・ カレンダー（スケジューラーの休日設定）
ASTERIA 3_HOME/data/schedule/holiday/*.holiday
→ DATA_HOME/flow/data/schedule/holiday
（ASMC でカレンダーを作成していない場合は不要です。）

４）フローに関連するデータ類をコピーする

※プロジェクトファイルの変換を行う場合、ここ以降の手順は後述の「3. ASTERIA WARP に対応したプロジェクトファイルに変換する手順」を参照してください。

ASTERIA 3 で使用していたユーザーのホームディレクトリにあるすべてのファイルを２）で作成したユーザーのホームディレクトリにコピーします。また、フローサービスを ASTERIA3 サーバーと異なるマシンにインストールした場合、ホームディレクトリ以外にフローで使用するすべてのデータファイルを同じファイルパスの場所にコピーします。

５）ASTERIA WARP で実行設定をインポートする

エクスポートしたファイル（fcs ファイル）をインポートするには、コマンドライン管理ツール flow-ctrl を使用します。flow-ctrl はインストーラーの中に含まれています。詳細については Web ドキュメントサイトから flow-ctrl リファレンスを参照してください。

1. WARP_HOME/flow/bin ディレクトリにある flow-ctrl.bat というバッチファイルをダブルクリックします。
2. 起動した画面から２）で作成したユーザー名とパスワードでログインします。
3. 以下のコマンドを実行します。
   ＞script［１でエクスポートした fcs ファイル名］

６）FTP の URL 通知を設定している場合

FTP の URL 通知設定は ASTERIA WARP では FTP 実行設定に変更されました。
URL 通知で実行していたフローがある場合は以下のようにフローを修正したうえで FTP 実行設定を行ってください。

1. フローに次の 2 つのフロー変数の追加します。
   ・ FilePath(String 型)　「公開」チェックボックスをオンにします
   ・ FileName(String 型)　「公開」チェックボックスをオンにします
2. MIMEDecode コンポーネントを削除します。
3. Start コンポーネントの出力ストリームを"Any"へ変更します。
4. Start コンポーネントと MIMEDecode の次に設定されているコンポーネント間を接続します。
   ※FTP にて転送されたファイル情報は、フロー変数の"FilePath"、"FileName"にて取得可能となります。

5. 修正したフローを FTP 実行設定します。

また、FTP のログインユーザーとは別のユーザーのフローを実行する場合は、実行したいフローを URL 実行設定します。次に、FTP のログインユーザーに新しく以下のようなフローを作成した後に FTP 実行設定を行ってください。

1. フローに次の 2 つのフロー変数の追加します。
   - FilePath(String 型)　「公開」チェックボックスをオンにします
   - FileName(String 型)　「公開」チェックボックスをオンにします
2. Start → Mapper → HttpPost → End となるフローを作成します。
   ここで、Mapper の出力ストリームの型は Text ストリームにして、フロー変数 FilePath を Mapper の出力ストリームとなるようにマッピングします。次に、HttpPost では「コネクションを使用」プロパティを「いいえ」にし、「URL」プロパティには URL 実行設定した別ユーザーのフローの URL を設定します。
3. このフローを FTP 実行設定します。

7）管理コンソールで必要事項を設定する

- ASTERIA3 で FTP を使用している場合、FTP 設定は ASTERIA3 と同じ設定をしてください。
- ASTERIA3 で SSL を使用している場合、再度、証明書をインポートしてください。

8）ASTERIA WARP を再起動する

フローサービスを停止し、再度フローサービスを起動します。

以上で移行のための手順は完了です。
管理コンソールから URL 実行設定やスケジュールからフローが実行されていることを確認してください。

2.2. ワークグループからの移行

ASTERIA WARP にはワークグループ機能がないため、ASTERIA 3 のワークグループで作成したシステムは通常ユーザーのシステムとして移行することになります。基本的な移行手順は通常ユーザーの場合と同じです。ここでは手順の中で注意が必要な点について注記します。

1）ASTERIA 3 から実行設定をエクスポートする
プロジェクトと実行設定のエクスポートはワークグループの asu でログインして行ってください。

2）ASTERIA WARP でアカウントを作成する
作成するユーザー名は任意ですが、ワークグループ名と同じにすることを推奨します。

4）フローに関連するデータ類をコピーする
ワークグループではフローは xfp ファイルではなく「_xpj」というフォルダに分割保存されています。

プロジェクトファイルを編集するための**変換を行わない**場合は分割形式で保存されているフローをあらかじめ xfp ファイルに変換しておく必要があります。分割形式のフローを通常の xfp ファイルとして保存するためには、ASTERIA 3 でワークグループの asu で接続して、フローの各プロジェクトを別ユーザーにコピーして保存してください。

プロジェクトファイルを編集するための**変換を行う**場合、変換ツール XFPConverter がワークグループに対応しているため上記手順は不要です。XFPConverter で指定する変換フォルダにはワークグループ

の asu のフォルダを指定してください。

3. ASTERIA WARP に対応したプロジェクトファイルに変換する手順

フローデザイナーでフローを編集するためにプロジェクトファイル（xfp ファイル）の変換を行う場合、前述「2.フロー実行環境の移行手順」の１）から３）の手順を行ったあとに本項の手順を行ってください。

①フローに関連するデータ類をコピーする

1. 一時的に使用する作業フォルダを任意の名前で作成します。
2. ASTERIA 3 で使用していたユーザーのホームディレクトリにあるすべてのプロジェクトファイル（xfp ファイル）を作業フォルダにコピーします。
3. プロジェクトファイル以外のファイルを２）で作成したユーザーのホームディレクトリにコピーします。
4. 異なるマシンにインストールした場合、ホームディレクトリ以外にフローで使用するすべてのデータファイルを同じファイルパスの場所にコピーします。

②プロジェクトファイルの変換

プロジェクトファイルを変換するためのツール「XFPConverter」がフローデザイナーのインストーラーに含まれています。XFPConverter は変換対象のプロジェクトファイルを置いたフォルダと出力先のフォルダを指定して実行するだけの簡単な GUI アプリケーションです。

1. フローデザイナーのインストールディレクトリにある convert.bat という名前のバッチファイルをダブルクリックして起動します。
2. 「変換フォルダ」にプロジェクトファイルをコピーした作業フォルダを指定します。
3. 「出力フォルダ」に２）で作成したユーザーのホームディレクトリを指定します。
4. 「実行」をクリックして変換します。

変換が終了すると出力フォルダには変換元と同名の xfp ファイルと xvar ファイルが生成されます。xvar ファイルは新しい機能である外部変数セットの定義ファイルです。プロジェクト内のプロジェクト定数、セッション変数は外部変数セットとして移行されます。プロジェクト定数、セッション変数を使用していない場合は生成されません。
ASTERIA WARP では ASTERIA 3 から様々な機能強化のための仕様変更がおこなわれているため、いくつかのケースでは XFPConverter での自動変換ができないことがあります。XFPConverter での制限事項については後述します。

③Velocity テンプレートの変更

プロジェクト定数、セッション変数を Velocity テンプレートから参照している場合、その参照キーワードを新しい機能である外部変数セットのキーワードに置換する必要があります。

$project → $exvar.<外部変数セット名>
$session → $exvar.<外部変数セット名>

外部変数セットは変換の際に自動的に生成されますが、その時の外部変数セット名はプロジェクト名と同じになります。つまり、Project1 のセッション変数「var1」の参照キーワードは「$exvar.Project1.var1」となります。

④ASTERIA WARP で実行設定をインポートする

エクスポートしたファイル（fcs ファイル）をインポートするには、コマンドライン管理ツール flow-ctrl を使用します。flow-ctrl はインストーラーの中に含まれています。詳細については管理コンソール ASMC のヘルプから flow-ctrl リファレンスを参照してください。

1. WARP_HOME/flow/bin ディレクトリにある flow-ctrl.bat というバッチファイルをダブルクリックします。
2. 起動した画面から２）で作成したユーザー名とパスワードでログインします。
3. 以下のコマンドを実行します。
   ＞script［１）でエクスポートした fcs ファイル名］

⑤FTP の URL 通知を設定している場合

FTP の URL 通知設定は ASTERIA WARP では FTP 実行設定に変更されました。
URL 通知で実行していたフローがある場合は以下のようにフローを修正したうえで FTP 実行設定を行ってください。

1. フローに次の 2 つのフロー変数の追加します。
   - FilePath(String 型)　「公開」チェックボックスをオンにします
   - FileName(String 型)　「公開」チェックボックスをオンにします
2. MIMEDecode コンポーネントを削除します。
3. Start コンポーネントの出力ストリームを"Any"へ変更します。
4. Start コンポーネントと MIMEDecode の次に設定されているコンポーネント間を接続します。
   ※FTP にて転送されたファイル情報は、フロー変数の"FilePath"、"FileName"にて取得可能となります。
5. 修正したフローを FTP 実行設定します。

また、FTP のログインユーザーとは別のユーザーのフローを実行する場合は、実行したいフローを URL 実行設定します。次に、FTP のログインユーザーに新しく以下のようなフローを作成した後に FTP 実行設定を行ってください。

1. フローに次の 2 つのフロー変数の追加します。
2. FilePath(String 型)　「公開」チェックボックスをオンにします
3. FileName(String 型)　「公開」チェックボックスをオンにします
4. Start → Mapper → HttpPost → End となるフローを作成します。
   ここで、Mapper の出力ストリームの型は Text ストリームにして、フロー変数 FilePath を Mapper の出力ストリームとなるようにマッピングします。次に、HttpPost では「コネクションを使用」プロパティを「いいえ」にし、「URL」プロパティには URL 実行設定した別ユーザーのフローの URL を設定します。
5. このフローを FTP 実行設定します。

⑥管理コンソールで必要事項を設定する

- ASTERIA3 で FTP を使用している場合、FTP 設定は ASTERIA3 と同じ設定をしてください。
- ASTERIA3 で SSL を使用している場合、再度、証明書をインポートしてください。

⑦ASTERIA WARP を再起動する

フローサービスを停止し、再度フローサービスを起動します。

以上で移行のための手順は完了です。
管理コンソールから URL 実行設定やスケジュールからフローが実行されていることを確認してください。

3.1. 変換ツール XFPConverter の制限事項

・　シナリオコンポーネントを使用している場合
シナリオ機能は廃止になったため、シナリオコンポーネントを使用したフローは ASTERIA WARP に移行することはできません。

・　SAP、Ariba、RosettaNet、ebXML、NewsML、Mainframe を使用している場合
これらのコンポーネントは現在 ASTERIA WARP に移植されていないため、移行することはできません。（移植の予定は未定です。）

・　条件式にマッパーで式を差し込んでいる場合
条件式（BranchStart コンポーネントの Condition プロパティなど）の中でプロジェクト定数やセッション変数を使用している場合、変換ツールにより自動的に外部変数セットを使用する式に変換されます。（例えば「$session.var1 = “aaa”」という式は「$exvar.Project1.var1 = “aaa”」と変換されます。）
ただし、条件式自体をマッパーで組み立ててマッピングしている場合は変換を行うことができません。プロジェクト定数やセッション変数を使用する式をマッパーで組み立ててマッピングしている場合は、変換後にフローデザイナーでその箇所を修正してください。

・　プロジェクト名またはフロー名に使用不可な文字を使用している場合
ASTERIA WARP ではプロジェクト名またはフロー名に以下の文字は使用できません。

!"#$%&'()=~^|¥@`+*;:{}[],.<>/?[タブ]

これらの文字が使用されている場合は名前を変更後にコンバートを行ってください。

・　同じ名前の引数とフロー変数を違う意味で使用している場合
ASTERIA WARP では引数（Start コンポーネントの Argument プロパティ）とフロー変数が統合されました。ASTERIA WARP では引数とは公開設定のフロー変数のことを指します。変換時には、引数名と同名で公開設定のフロー変数名が定義されます。

引数とフロー変数に同じ名前が定義されていた場合、変換ツールではそれらは同じ変数を扱っているものとしてマージしています。変換元のフローで、Start コンポーネントで定義した引数をその直後のマッパーでフロー変数に移している場合、マッピングした先のフロー変数と引数が同じ名前で定義されているときには最初に同じフロー変数への代入（var1 = var1）が行われるだけなので問題ありません。

しかし、同じ名前の引数とフロー変数を異なる意味で使用している場合はマージした結果が不正な代入式となる可能性があるので、変換後にフローデザイナーでフロー変数の使用箇所を確認してください。

・　プロジェクト定数とセッション変数で同じ変数名を使用している場合
プロジェクト定数とセッション変数は ASTERIA WARP では外部変数セットに変換されますが、外部変数セットでは定数とセッション変数に同じ名前を使用することができません。（これは最初にリクエスト変数として定義した変数を後からセッション変数に変更するなど変数種別の変更を容易にするためです。）このため、プロジェクト定数とセッション変数で同じ名前を使用している場合は正しく変換できません。あらかじめ変換前のプロジェクトファイルのプロジェクト定数名を別の変数名に変更する前処理を行うことで正しい変換を行うことができるようになりますが、このような前処理は変換ツールには含まれていないので別途御相談ください。

・　コンポーネントプロパティのファイルパス指定で相対パスを指定している場合
ASTERIA 3 ではファイルパスを指定するプロパティで相対パスを指定した場合、「..」を使用することで、相対パスの起点となるディレクトリよりも上位のディレクトリを指定することができましたが、ASTERIA WARP では相対パス指定の場合は、セキュリティ上の観点から、相対パスの起点となるディレクトリよりも上位のディレクトリは指定することができなくなりました。

このような相対パスが指定されている場合は、フローの変換後に、相対パスの指定を絶対パス指定に変更するなど、ファイルパスの指定を見直してください。

・ 複数の終了コンポーネントで CloseSession プロパティが異なる場合
ASTERIA 3 では各終了コンポーネントに CloseSession プロパティがありましたが、ASTERIA WARP ではこのプロパティはフローの「セッション」プロパティに変更されました。

このため、フロー内で分岐して一方の終了コンポイーネントの CloseSession プロパティが True、他方の終了コンポーネントの CloseSession プロパティが False となっているような場合、正しく変換できません。変換後にフローデザイナーで該当箇所の終了コンポーネントに NextFlow を使って「セッション」プロパティを替えた別のフローを呼び出すなどの修正を行ってください。

・ Mapper の出力側のストリームやフィールドに複数のリンク線をつないでいる場合
ASTERIA 3 では出力側のストリームやフィールドに複数のリンク線をつなぐことができましたが、ASTERIA WARP では 1 つしかリンク線をつなぐことができなくなりました。このため、複数のリンク線をつないでいる場合には NullCheck 関数などを使用してリンク線を 1 つにする必要があります。

・ サブフローの引数に対するフロー変数の自動マッピング機能を使用している場合
ASTERIA 3 でサブフローに対して値を渡す場合、サブフローコンポーネントのプロパティとして表示されるフロー変数を使う方法があります。

これとは別に、サブフローで定義した引数に呼出し元フローのフロー変数が自動的に設定されて値を渡せる方法がありましたが、ASTERIA WARP では引数とフロー変数が統合されたため、この仕様は廃止されました。この方法を使用している場合は変換後にフローデザイナーでサブフローの呼び出し箇所でフロー変数をサブフローコンポーネントのパラメータにマッピングするように修正してください。

（Exception フローの場合は ASTERIA 3 の場合と同様に呼出し元フローのフロー変数が Exception フローの公開設定のフロー変数に自動的に設定されます。）

・ End コンポーネントで終了するフローをサブフローとして呼び出している場合
変換ツールにより変換されたフローでは、サブフローの入力ストリームの型が出力ストリームに定義されずに、後続のマッパー等でマッピングできないエラーが発生することがあります。このような場合には、デザイナーで SubFlow コンポーネントを右クリックし、「サブフロー情報の更新」を実行する必要があります。

・ サブフロー内でエラーが発生したときに呼び出す Exception フローが End コンポーネントで終了している場合
ASTERIA3 では、Exception フローが終了するとサブフローが終了したことになり、メインフローの処理は継続されます。ASTEIRA WARP では、Exception フローが終了するとサブフロー終了後にメインフローもそのまま終了します。ASTERIA3 と同じ動作をするようにするには、End コンポーネントの代わりに ExceptionReturn コンポーネントを使用します。

・ XPathString、XpathNodeSet、XMLUpdate、XMLMerge コンポーネントでプロジェクト定数やセッション変数を参照している場合
ASTERIA 3 ではこれらのコンポーネントの Xpath 式の中でプロジェクト定数とセッション変数が参照できましたが、ASTERIA WARP の外部変数セットは Xpath 式からは参照できません。変換後にフローデザイナーで該当箇所をフロー変数やローカル変数を使用するように変更してください。

・EDIFACTEncoder コンポーネントで入力ストリームを定義せずに使用している場合
ASTERIA WARP では EDIFACTEncoder コンポーネントを使用する場合、入力ストリーム定義が必須となりました。そのため、入力ストリーム定義を持たない EDIFACTEncoder コンポーネントは正しく変換できません。

・ BranchByComponentProperty コンポーネントで「プロパティ名」プロパティの値を直前または直後のマッパーで設定、参照している場合

ASTEIRA WARP では、BranchByComponentProperty コンポーネントの「プロパティ名」プロパティはマッパーでのマッピングはできなくなりました。「プロパティ名」プロパティを再度選択しなおすか、BranchStart コンポーネントに置き換えてください。

・ BranchByComponentProperty コンポーネントで「プロパティ名」プロパティに直前のサブフローのパラメータを設定している場合

ASTERIA3 ではパラメータ名だけを指定しましたが、ASTERIA WARP では「パラメータ.パラメーター名」という形式に変わりました。変換ツールでの変換後に BranchByComponentProperty コンポーネントの「プロパティ名」プロパティを再度設定し直してください。

・ BranchByComponentProperty コンポーネントの「プロパティ名」プロパティで次のプロパティを設定している場合
- FileList コンポーネントの FileCount プロパティ
- FTPFileList コンポーネントの FileCount プロパティ
- RecordFilter コンポーネントの RecordCount プロパティ
- RecordJoin コンポーネントの RecordCount プロパティ
- RecordMatch コンポーネントの RecordCount プロパティ
- RecordSort コンポーネントの RecordCount プロパティ
- RecordSQL コンポーネントの RecordCount プロパティ
- RDB(Get)コンポーネントの RecordCount プロパティ

上記のプロパティはASTERIA WARP ではストリーム変数$stream.RecordCountに変更になりました。BranchByComponentProperty コンポーネントの「プロパティ名」プロパティに上記のプロパティを設定している場合、変換ツールにより「条件式」プロパティに$stream.RecordCount を使用した BranchStart コンポーネントに置き換えられます。ただし、上記のコンポーネントと BranchByComponentProperty コンポーネントの間に BranchByException 等のコンポーネントが配置されている場合には、変換ツールによる BranchStart コンポーネントへの置き換えが行われません。このような場合には、変換ツールによるフローの変換後に BranchByComponentProperty コンポーネントを BranchStart コンポーネントへ置き換えて、「条件式」プロパティを$stream.RecordCount を使用した条件式に再設定する必要があります。

3.2. 変換ツールのその他注意事項

・ マッパーのレイヤーは移行されません

ASTERIA 3 のマッパーで複数レイヤーを使用してマッピングしていた場合、1 つのレイヤーに移行されます。（実行には支障がありません。）

・ リンク線は初期化されます

フローのリンク線の種類は変換元と同じです。リンク線の位置を連結したあとに移動している状態だったときには初期状態に戻ります。また、ASTERIA3 デザイナーとフローデザイナーでは、曲線のカーブの形状が異なります。

マッパーのリンク線の種類は変換後すべて直線になります。

・ 終了コンポーネントの変換について

ASTERIA3 では終了コンポーネントは

- End
- EndResponse
- Break
- Abort

の 4 つのコンポーネントでしたが、WARP では機能毎に細かく分割され以下の 7 つとなっています。

- End
- EndResponse
- HttpEnd
- NextFlow
- ExceptionReturn
- Break
- Exception

これに伴いいくつかのプロパティも廃止されておりコンバータでは以下のように変換されます。

- EndResponse コンポーネントで NextFlow プロパティに値が設定(またはマッピング)されていた場合
NextFlow コンポーネントに変換されます。
EndResponse コンポーネントの NextFlow プロパティは廃止されました。

- End/EndResponse コンポーネントで Location プロパティに値が設定(またはマッピング)されていた場合
HttpEnd コンポーネントに変換されます。
End/EndEndResponse コンポーネントの Location プロパティは廃止されました。

- EndResponse コンポーネントで ExceptionReturn プロパティの値が「EndOfRequest」以外の場合
ExceptionReturn コンポーネントに変換されます。
EndResponse コンポーネントの ExceptionReturn プロパティは廃止されました。

コンポーネントの変換は上記の順番で実行されます。
ASTERIA3 では定義上、NextFlow プロパティ、Location プロパティ、ExceptionReturn プロパティのすべてに値を設定することができますが、WARP への変換時には NextFlow プロパティに値があれば NextFlow コンポーネントに変換され、Location プロパティ、ExceptionReturn プロパティは無視されます。

- Abort コンポーネントを使用していた場合
WARP では Abort コンポーネントは廃止され Exception コンポーネントに置き換えられましたが、
コンバート時には Exception コンポーネントには変換されずそのまま Abort コンポーネントとなります。

上記以外で ASTERIA3 で廃止されたプロパティ、または設定値が指定されていた場合はそのプロパティがそのまま移行され、設定どおりに動作します。
ただし、それは本来 WARP では設定できない内容ですので可能な限りフローを修正してください。

ケースとしては以下の場合があります。

- End/EndResponse/Break コンポーネントの Transaction プロパティに「DoNothing」を指定していた場合
ASTERIA3 では Transaction プロパティの設定値は「Commit/Rollback/DoNothing」の 3 択でしたが、WARP では NextFlow コンポーネント以外の終了コンポーネントでの設定値は「コミット／ロールバック」の 2 択に変更されています。
本来 NextFlow で次処理を続けて実行する場合以外はトランザクションをコミットもロールバックもしないという選択はあり得ないので、ロジックを見直してください。

- Break コンポーネントで ExceptionReturn プロパティが指定されていた場合
WARP では Break コンポーネントから直接 ExceptionReturn する設定はできなくなりました。
LoopEnd と組み合わせて ExceptionReturn コンポーネントで復帰するように修正してください。

- Break コンポーネントで NextFlow プロパティが指定されていた場合

WARP では Break コンポーネントから直接 NextFlow を実行する設定はできなくなりました。
LoopEnd と組み合わせて NextFlow コンポーネントで次フローを指定するように修正してください。

- Break コンポーネントで Location プロパティが指定されていた場合
WARP では Break コンポーネントから直接 HTTP 実行時のリダイレクトの設定はできなくなりました。
LoopEnd と組み合わせて HttpEnd コンポーネントでリダイレクトを指定するように修正してください。

改版履歴

版数	日付	内容
第 1 版	2006/10/23	新規作成
第 2 版	2006/12/11	FTP 実行設定移行手順、XFPConverter での EDIFACTEncoder コンポーネントの制限事項を追加
第 3 版	2008/06/18	制限事項にプロジェクト名、フロー名の使用可能文字に関する制限を追加
第 4 版	2008/08/07	制限事項に相対パスの指定に関する制限を追加
第 5 版	2009/02/20	制限事項で「XPathString、XpathNodeSet コンポーネントでプロジェクト定数やセッション変数を参照している場合」の対象に XMLUpdate、XMLMerge コンポーネントを追加
第 6 版	2010/08/23	制限事項にサブフロー内でエラーが発生したときに呼び出す Exception フローが End コンポーネントで終了している場合を追加
第 7 版	2011/03/29	制限事項に BranchByComponentProperty コンポーネントの「プロパティ名」プロパティの値を直前または直後のマッパーで設定、参照している場合を追加
第 8 版	2012/01/19	FTP 実行設定以降手順に別ユーザーのフローを実行する場合の手順を追加 制限事項に BranchByComponentProperty コンポーネントで「プロパティ名」プロパティに直前のサブフローのパラメータを設定している場合を追加 変換ツールのその他の注意事項に「終了コンポーネントの変換について」を追加
第 9 版	2013/01/16	制限事項に End コンポーネントで終了するフローをサブフローとして呼び出している場合と Mapper の出力側のストリームやフィールドに複数のリンク線をつないでいる場合を追加
第 10 版	2013/01/21	制限事項に BranchByComponentProperty コンポーネントの「プロパティ名」プロパティで次のプロパティを設定している場合を追加