



Peregrine | AssetCenter
マイグレーション

© Copyright 2002 Peregrine Systems, Inc.

All rights reserved.

本書に記載されている情報は、Peregrine Systems, Incorporatedが所有し、Peregrine Systems, Inc.の書面による許可なく使用または開示することはできません。本書の一部または全部を、Peregrine Systems, Inc.の事前の書面による許可なく無断で複製することを禁じます。本書に記載されている商品名は、該当する各社の商標または登録商標です。

Peregrine Systems ®およびAssetCenter ®は、Peregrine Systems, Inc.の登録商標です。

本書で説明されているソフトウェアは、Peregrine Systems, Inc.とエンドユーザ間で締結されるライセンス契約に基づいて提供されます。契約の条項に従って、ソフトウェアを使用する必要があります。Peregrine Systems, Inc.は、本書の内容については一切の責任を負いかねます。また、本書の内容が予告なく変更されることもあります。本書の最終バージョンの日付を確認するには、Peregrine Systems, Inc.のカスタマサポートまでお問合せください。

デモ用データベースと本書の例に使用されている団体名および個人名は架空のものであり、本ソフトウェアの使用方法を説明するためのものです。現在、過去を問わず、実在する団体や個人とのいかなる類似もまったくの偶然によるものです。

本製品に関する技術情報の請求、またはライセンスをお持ちの製品に関するマニュアル類の請求については、Peregrine Systemsのカスタマサポート (support@peregrine.com) までお寄せください。

本マニュアルに関するご意見やご要望は、Peregrine Systems, Inc.の出版部 (doc_comments@peregrine.com) までお寄せください。

本書の内容は、ライセンス契約に基づくプログラムのバージョン4.2.0に適用されます。

AssetCenter

Peregrine Systems, Inc.
Worldwide Corporate Campus and Executive Briefing Center
3611 Valley Centre Drive San Diego, CA 92130
Tel 800.638.5231 or 858.481.5000
Fax 858.481.1751
www.peregrine.com



目次

はじめに	11
AssetCenter 4.1.0またはそれ以降のバージョンを移行する場合	11
マイグレーションを行う理由	11
マイグレーション作業の内容	11
マイグレーションの対象ユーザ	12
必須要素	12
本マニュアルの使用方法	12
AssetCenter 4.0.0またはそれ以前のバージョンを移行する場合	13
マイグレーションを行う理由	13
マイグレーション作業の内容	13
マイグレーションの対象ユーザ	14
必須要素	14
本マニュアルの使用方法	14
1. サポートされる環境	17
2. AssetCenter 4.1.0またはそれ以降のバージョンを移行する	19
方法1「部分」	19
方法2「完全」	20
3. AssetCenter 4.0.0またはそれ以前のバージョンを移行する - 概要	25

4. 段階を追ってマイグレーションを実行する - 準備段階（本番データベース）	33
事前分析	33
マイグレーションプロジェクトを開始する	34
ユーザおよびヘルプデスクエンジニアの教育	35
変換用コンピュータを準備する	35
DBMSサーバを準備する	37
5. 段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）	39
1 手順1 - 旧フォーマットの本番データベースの整合性を検証する	40
2 手順2 - 旧フォーマットの本番データベースを手動調整する	41
調整作業（旧フォーマットの本番データベースの全バージョン用）	41
調整作業（旧フォーマットの本番データベースが4.0.0バージョンより前の場合）	42
3 手順3 - 旧フォーマットの本番データベースの構造の変更事項を伝達する	51
4 手順4 - 旧フォーマットの本番データベースのコピーを作成する	56
5 手順5 - 旧フォーマットの本番データベースをシミュレーション用データベースに変換する	57
変換ファイル「migration.xml」の調整	58
旧フォーマットの本番データベースをシミュレーション用データベースに変換する	59
変換に関する情報	61
6 4.2.0フォーマットの本番データベースの整合性を検証する	70
7 4.2.0フォーマットの本番データベースの有効性を確認する	71
8 手順8 - 旧フォーマットの本番データベースでの権限の一部を制限する	72
9 手順9 - 手動で変換するアプリケーションデータを出力する	73
手動で変換するアプリケーションデータを処理する	76
10 手順10 - アプリケーションデータを検証し、修正する	77
11 手順11 - 修正されたアプリケーションデータを復元する	88
12 手順12 - 4.2.0フォーマットの本番データベースの整合性を検証する	90
13 手順13 - 復元されたアプリケーションデータを検証する	91
14 手順14 - 外部アプリケーションとの統合を調整する	92

6. 段階を追ってマイグレーションを実行する - 最終変換（移行データベース）	93
15 手順15 - 旧フォーマットの本番データベースの整合性を検証する	93
16 手順16 - 旧フォーマットの本番データベースをブロックしコピーを作成する	94
17 手順17 - 旧フォーマットの移行データベースを変換する	94
18 手順18 - 手動で変換されたアプリケーションデータを復元する	95
19 手順19 - 4.2.0フォーマットの移行データベースで整合性を検証する	95
20 手順20 - 4.2.0フォーマットの移行データベースを完成させる	96
完成作業（旧フォーマットの本番データベースの全バージョン用）	96
完成作業（旧フォーマットの本番データベースが4.0.0バージョンより前の場合）	106
7. 段階を追ってマイグレーションを実行する - 最終段階	113
AssetCenterプログラムを更新する	113
AssetCenter Serverを管理用コンピュータにインストールする	114
4.2.0フォーマットの移行データベースのAssetCenterキャッシュを削除する	114
AssetCenterプログラムを更新する	114
AssetCenterが正常に起動することを確認する	117
古い接続を削除して、新しい接続を作成する	117
必要に応じてクライアントでAssetCenterのカスタマイズ設定を変更する	117
4.2.0フォーマットの移行データベースを実際に使用する	117
8. 用語解説	119
マイグレーション	119
AssetCenterプログラムを更新する	119
旧フォーマットの本番データベースを変換する	120
変換ファイル	120
変換用コンピュータ	120
本番データベース	120
トリガ	121
データ	121
アプリケーションデータ	121
データベースの構造	122
9. 参考情報	123
変換ファイル「migration.xml」の調整	123
警告	123

参考情報	124
変換ファイルの役割	124
変換規則	124
変換ファイルのシンタックス	125
特殊文字を使用する	132
旧テーブルのフィールドを新規の複数テーブルへ配分する	134
任意管理項目をフィールドへ転送する	134
手動変換用アプリケーションデータを格納するフィールドを変換する	136
結合を使用する	137
外部キーを使用する	137
ソーステーブルを複数のターゲットテーブルに配分する	138
数値フィールドをテキスト文字列へ変換する	138
アプリケーションデータを手動で変換する	138
変換ファイルから生成されるSQL命令	139
変換ファイルを使用前に検証する	139
特定のテーブルに関連していない資産の転送	140
標準データベースの構造と旧バージョンの構造の相違	140
手動で変換するアプリケーションデータ	143
AssetCenter Script Analyzerを使って変更できるアプリケーションデータ	143
確認するその他のアプリケーションデータ	146
旧フォーマットの本番データベースの伝達された構造パラメータ	147
その他のマニュアル	148

図の一覧表

2.1. 変換 - プロセス	21
3.1. 変換 - プロセス	27
5.1. 構造の変更事項の伝達 - プロセス	53
5.2. サンプルデータの処理 - プロセス	75
5.3. AssetCenter Script Analyzer - 「*.xml」ファイルの分析用ウィンドウ	83
5.4. AssetCenter Script Analyzer - スクリプトの分析用ウィンドウ	86

表の一覧表

5.1. 文字「^」を含んではならないフィールド - 一覧	44
5.2. 文字「/」を含んではならないフィールド - 一覧	50
5.3. AssetCenter Script Analyzer - メニュー	81
9.1. 手動変換するアプリケーションデータ - 一覧	143
9.2. 旧フォーマットの本番データベースの構造パラメータ - 一覧	147
9.3. その他のマニュアル - 一覧	148

AssetCenter 4.1.0またはそれ以降のバージョンを移行する場合

マイグレーションを行う理由

旧バージョン（4.1.0またはそれ以降）と4.2.0バージョン間では、AssetCenterデータベースが以下の点で少々変更されています。

- データベースの構造：
特定の構造パラメータ（例：デフォルト値）が変更されました。
ただし、テーブル、フィールドおよびリンクに関しては、一切追加 / 削除されていません。
- プログラムが少々変更されました。

このためAssetCenter 4.2.0を使用するには、マイグレーションを実行する必要があります。

マイグレーション作業の内容

マイグレーション作業では、以下のタスクを実行します。

- 1 AssetCenterプログラムをバージョン4.2.0へ更新する。
- 2 バージョン4.2.0のシステムデータを本番データベースへインポートする。
- 3 場合に応じて、**旧フォーマットの本番データベース**の構造パラメータ（例：デフォルト値）を、4.2.0フォーマットに変換する。

マイグレーションの対象ユーザ

AssetCenterの4.1.0以降のバージョンを使用する会社が、バージョン4.2.0に移行する場合、マイグレーションを行う必要があります。

以下のエンジニアが、マイグレーションを実施します。

- AssetCenterデータベースの管理を担当するエンジニア
- AssetCenterのインストールを担当するエンジニア
- AssetCenterの導入を担当するエンジニア

必須要素

マイグレーションは、以下の要素を要する簡単なプロセスです。

- AssetCenterの知識（インストール、管理）
- 準備
- 技術的能力：データベース管理
- メソッド

本マニュアルの使用方法

ヒント:

本マニュアルを読む前に、以下のAssetCenter 4.2.0の付属マニュアルをお読みください。

- マニュアル『インストール』
- リリースノート
- readme.txt

ヒント:

本マニュアルで通読する必要はありません。以下の章のみをお読みください。

「サポートされる環境」の章

この章では、マイグレーションがサポートする環境が説明されています。使用中の環境がサポートされるかどうかを確認してください。

「AssetCenter 4.1.0またはそれ以降のバージョンを移行する」の章

ここでは、4.2.0バージョンへのマイグレーションの各段階が説明されています。

「段階を追ってマイグレーションを実行する - 最終段階」の章、「AssetCenter プログラムを更新する」の節

この節では、AssetCenterプログラムを更新する方法が説明されています。

「用語解説」の章

この章では、マイグレーションに関連する主要用語が説明されています。この用語解説を参照して、本マニュアル内の用語の理解を深めてください。

AssetCenter 4.0.0またはそれ以前のバージョンを移行する場合

マイグレーションを行う理由

バージョン4では、AssetCenterの構造が大幅に変更されました。

- データベースの標準構造（テーブル、フィールド、リンク、インデックス）が大幅に変更されました。
例えば、旧バージョンの「カテゴリ/製品/資産」という構造が、新バージョンでは「属性/モデル/資産、ロット、ポートフォリオ品目およびカタログ製品」という構造を取ります。
- 新しい機能が追加されました。

上記の理由から、旧バージョンのAssetCenterをバージョン4.2.0に体系的にマイグレーションする必要性が生じます。

マイグレーション作業の内容

マイグレーション作業では、以下のタスクを実行します。

- 旧フォーマットの本番データベースを4.2.0フォーマットに変換する（構造と内容）。
- AssetCenterプログラムをバージョン4.2.0へ更新する。

マイグレーションの対象ユーザ

AssetCenterの4.2.0バージョンよりも前のバージョンを使用する会社で、バージョン4.2.0に移行する場合、マイグレーションを行う必要があります。

以下のエンジニアが、マイグレーションを実施します。

- AssetCenterデータベースの管理を担当するエンジニア
- AssetCenterのインストールを担当するエンジニア
- AssetCenterの導入を担当するエンジニア

必須要素

マイグレーションは、以下の要素を要する複雑なプロセスです。

- AssetCenterの旧バージョンとバージョン4.2.0に関する高度な知識（インストール、設定、データベース構造、作動、管理、外部アプリケーションとの結合）
- 準備
- 技術的能力：SQL、データベース管理
- メソッド
- 時間
- リソース

本マニュアルの使用方法

ヒント:

本マニュアルを通読する前に、以下のAssetCenter 4.2.0の付属マニュアルをお読みください。

- マニュアル『インストール』
- リリースノート
- readme.txt
- マニュアル『バージョン3.xと4.1.0の相違点』

 **ヒント:**

本マニュアルは、以下の順番に従って通読するようお勧めします。

「サポートされる環境」の章

この章では、マイグレーションがサポートする環境が説明されています。使用中の環境がサポートされるかどうかを確認してください。

「AssetCenter 4.0.0またはそれ以前のバージョンを移行する - 概要」の章

この章では、マイグレーションの概要が説明されています。ここでマイグレーションの全体像を把握してください。

「段階を追ってマイグレーションを実行する - 準備段階（本番データベース）」の章**「段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）」の章****「段階を追ってマイグレーションを実行する - 最終変換（移行データベース）」の章****「段階を追ってマイグレーションを実行する - 最終段階」の章**

これらの章はマイグレーションの各段階を説明しています。マイグレーションのプロセスの全体像を把握するためには、各章を通読してください。ここで説明されている順番通りに、詳細に注意を払いながら各段階の作業を実行します。

「用語解説」の章

この章では、マイグレーションに関連する主要用語が説明されています。この用語解説を参照して、本マニュアル内の用語の理解を深めてください。

「参考情報」の章

ここでは、参考情報が記載されています。
また、高度な情報や補足情報も記載されています。

1 | サポートされる環境

オペレーティングシステムとDBMS

マイグレーションでは、AssetCenterがサポートするすべてのオペレーティングシステムとDBMSが使用可能です（SQL Anywhereランタイムを除きます）。サポートされるオペレーティングシステムとDBMSの一覧については、Webサイト<http://support.peregrine.com>の動作環境の表を参照してください。

AssetCenter本番データベース

マイグレーションは、次のデータベースの変換をサポートします。

- AssetCenterバージョン3.01以降の、サービスパックも含めたすべてのバージョン
本番データベースが3.01より前のバージョンの場合は、データベースをまずバージョン3.02のフォーマットに変換します。
データベースをバージョン3.02に変換する方法については、以下のマニュアルを参照してください。
 - マニュアル『AssetCenter - バージョン3.0 - インストールおよびアップグレードバージョン』、「AssetCenterのアップグレード」の章
 - バージョン3.02のReadme.txt、「はじめに」の節
- AssetCenter Cable and Circuit 3.10

🚩 重要項目:

マイグレーション時に、ソース言語とターゲット言語は同一でなければなりません。

例：AssetCenter 3.6.0ドイツ語版から、4.2.0英語版へのマイグレーションは不可能です。

DBMSサーバのディスク領域

⚠ 警告:

AssetCenter 4.1.0またはそれ以降のバージョンを移行する場合、この節の情報は関係ありません。

DBMSサーバが旧フォーマットの本番データベースへ割り当てるディスク領域は、旧フォーマットの本番データベースのサイズの2倍以上でなければなりません。

2 | AssetCenter 4.1.0またはそれ以降のバージョンを移行する

マイグレーションを実行する方法としては、以下の2とおりの方法があります。

- 旧フォーマットの本番データベースの構造パラメータを変換しない場合は、方法1「部分」が適しています。
- 旧フォーマットの本番データベースの構造パラメータを変換する場合は、方法2「完全」が適しています。

「構造パラメータ」の定義については、本マニュアルの「[参考情報](#)」の章、「[旧フォーマットの本番データベースの伝達された構造パラメータ](#)」の節を参照してください。

構造パラメータを変更するべきかどうかを検討するには、旧フォーマットの本番データベースの構造と、4.2.0フォーマットの本番データベースの構造の相違点を分析する相違ファイルを参照してください。

相違ファイルの格納先や使用方法については、本マニュアルの「[参考情報](#)」の章、「[標準データベースの構造と旧バージョンの構造の相違](#)」の節を参照してください。

方法1 「部分」

- 1 旧フォーマットの本番データベースのバックアップコピーを作成します。

- 2 本マニュアルの「**段階を追ってマイグレーションを実行する - 最終変換（移行データベース）**」の章の「**手順16- 旧フォーマットの本番データベースをブロックコピーを作成する**」の節で説明されている通りに、旧フォーマットの本番データベースへのアクセスをブロックします。
- 3 「**段階を追ってマイグレーションを実行する - 最終段階**」の章の「**AssetCenterプログラムを更新する**」の節で説明されている通りに、AssetCenterプログラムを更新します。
- 4 旧フォーマットの本番データベースのDBMSがDB2である場合、「db2udb72.sql」ファイルのSQL命令を旧フォーマットの本番データベースに対して実行します。一般的に「db2udb72.sql」ファイルへのパスは、「C:\Program Files\Peregrine\AssetCenter\migration\db2udb72」です。
- 5 **バージョン4.2.0のシステムデータをインポート**します。
 - 1 AssetCenterを起動します。
 - 2 ブロックされた旧フォーマットの本番データベースに接続します（**[ファイル/データベースに接続]**メニュー）。
 - 3 **[ファイル/インポート]**メニューを選択してから、**[スクリプトの実行]**オプションを選択します。
 - 4 スクリプト「upgrade.lst」を選択します。一般的にこのスクリプトは、「C:\Program Files\Peregrine\AssetCenter\migration」フォルダに格納されています。
 - 5 **[インポート]**をクリックします。
 - 6 **[閉じる]**をクリックします。
 - 7 このようにして作成されたデータベースを**4.2.0フォーマットの本番データベース**と呼びます。
- 6 4.2.0フォーマットの本番データベース上でAssetCenter Serverを起動します。
- 7 4.2.0フォーマットの本番データベースにアクセスする外部プログラムを、再起動します。
- 8 データベースが使用可能であることをユーザに通知します。

 **警告:**

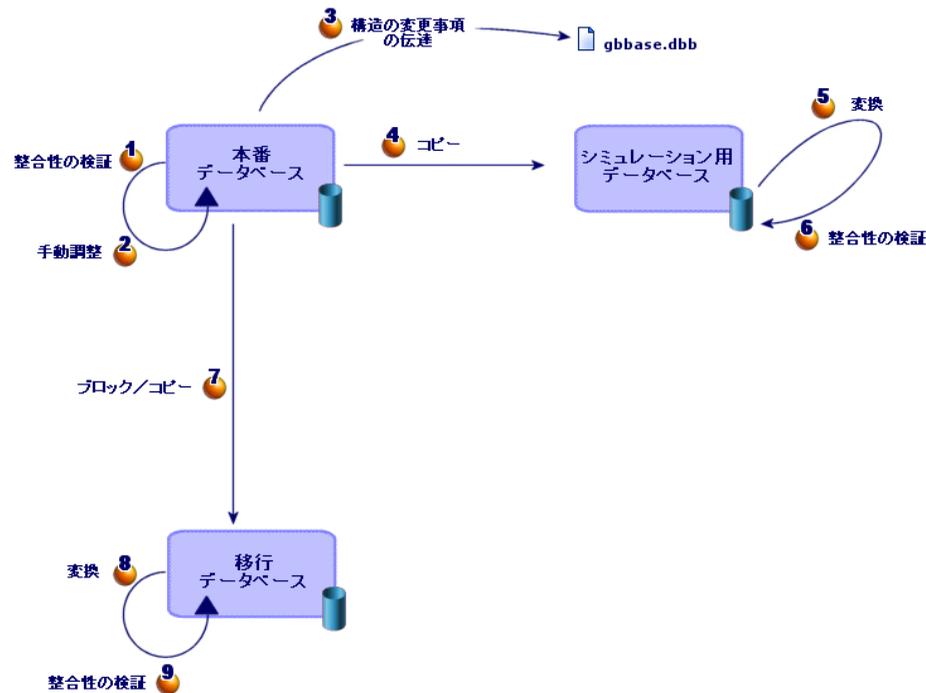
方法1「部分」でAssetCenter 4.1.0またはそれ以降のバージョンを移行する場合、本節から先を読む必要はありません。

方法2「完全」

変換の実行方法

変換プロセスの概要は以下の通りです。

図 2.1. 変換 - プロセス



以下の手順に従って変換を実行します。

1 シミュレーション用データベース上で変換のシミュレーションを実行する。

1 AssetCenter Database Administratorを使って旧フォーマットの本番データベースの整合性を検証します。

本マニュアルの「段階を追ってマイグレーションを実行する - シミュレーション (シミュレーション用データベース)」章の「手順1 - 旧フォーマットの本番データベースの整合性を検証する」の節の指示に従って、作業を進めてください。

2 必要に応じて、AssetCenterを使って旧フォーマットの本番データベースを手動で調整します。

本マニュアルの「段階を追ってマイグレーションを実行する - シミュレーション (シミュレーション用データベース)」章、「手順2 - 旧フォーマットの本番データベースを手動調整する」の節、「調整作業 (旧フォーマットの本番データベースの全バージョン用)」の項の指示に従って、作業を進めてください。

- 3** 旧フォーマットの本番データベースに実行した構造の変更事項を、「標準 4.2.0 gbbase.dbb」データベース記述ファイルへ伝達します。

本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」の章の「[手順3-旧フォーマットの本番データベースの構造の変更事項を伝達する](#)」の節の指示に従って、作業を進めてください。
- 4** 旧フォーマットの本番データベースのコピーを作成します (このコピーは「シミュレーション用データベース」と呼ばれます)。シミュレーション用データベースで変換シミュレーションを試行している間、他のユーザは旧フォーマットの本番データベースを通常どおり使用することができます。

本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」の章の「[手順4-旧フォーマットの本番データベースのコピーを作成する](#)」の節の指示に従って、作業を進めてください。
- 5** AssetCenter Database Administratorを使って、旧フォーマットのシミュレーション用データベースを変換します。

これにより、ユーザ企業が実行した構造変更を保持しつつ、旧フォーマットのシミュレーション用データベースの構造パラメータが変換されます。

本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」章、「[手順5-旧フォーマットのシミュレーション用データベースを変換する](#)」の節、「???」の項の指示に従って、作業を進めてください。
- 6** AssetCenter Database Administratorを使って、4.2.0フォーマットのシミュレーション用データベースの整合性を検証します。

この操作では、変換がシミュレーション用データベースを破損していないかどうかを確認します。

本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」章の「[4.2.0フォーマットのシミュレーション用データベースの整合性を検証する](#)」の節の指示に従って、作業を進めてください。
- 2** 旧フォーマットの本番データベースの2番目のコピー (「移行データベース」) を実際に変換する。
- 7** 旧フォーマットの本番データベースをブロックし、コピー「移行データベース」を作成します。

本マニュアルの「[段階を追ってマイグレーションを実行する - 最終変換 \(移行データベース\)](#)」章の「[手順16-旧フォーマットの本番データベースをブロックしコピーを作成する](#)」の節の指示に従って、作業を進めてください。
- 8** AssetCenter Database Administratorを使って、旧フォーマットの移行データベースを変換します。

本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」章、「[手順5 - 旧フォーマットのシミュレーション用データベースを変換する](#)」の節、「[???](#)」の項の指示に従って、作業を進めてください。

旧フォーマットのシミュレーション用データベースへ接続するかわりに、旧フォーマットの移行データベースに接続します。

 AssetCenter Database Administratorを使って、4.2.0フォーマットの移行データベースの整合性を検証します。

本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」章の「[4.2.0フォーマットのシミュレーション用データベースの整合性を検証する](#)」の節の指示に従って、作業を進めてください。

旧フォーマットの本番データベースへ接続するかわりに、4.2.0フォーマットの移行データベースに接続します。

- 3 必要に応じて、ストアードプロシージャup_GetCounterValを変更します。
本マニュアルの「[段階を追ってマイグレーションを実行する - 最終変換 \(移行データベース\)](#)」章、「[手順20 - 4.2.0フォーマットの移行データベースを完成させる](#)」の節、「[ストアードプロシージャup_GetCounterValの変更](#)」の項の指示に従って、作業を進めてください。
- 4 「[段階を追ってマイグレーションを実行する - 最終段階](#)」の章の「[AssetCenterプログラムを更新する](#)」の節で説明されている通りに、AssetCenterプログラムを更新します。
- 5 4.2.0フォーマットの本番データベース上でAssetCenter Serverを起動します。
- 6 4.2.0フォーマットの本番データベースにアクセスする外部プログラムを再起動します。
- 7 データベースが使用可能であることをユーザに通知します。

 **警告:**

方法2「完全」でAssetCenter 4.1.0またはそれ以降のバージョンを移行する場合、本節から先を読む必要はありません。

3 | AssetCenter 4.0.0またはそれ以前のバージョンを移行する - 概要

マイグレーション作業の内容

マイグレーションとは、AssetCenterの旧バージョンからバージョン4.2.0に移行するために実行する以下の操作の集まりのことを指します。

- 旧フォーマットの本番データベースを変換して（構造と内容）、AssetCenterのバージョン4.2.0と互換性があるようにする。
- 管理用コンピュータとユーザコンピュータでAssetCenterプログラムをバージョン4.2.0にアップグレードする。

データベースの変換は複雑な作業であるため、本章では概要のみを説明します。プログラムの更新は一般的な操作であるため、ここでは概要の説明は省きます。

変換作業の内容

データベースを変換するには、以下の操作を行います。

- データベース構造をAssetCenterのバージョン4.2.0と同じ構造に変更する。
- できる限り元のデータを維持する。
- データベース構造の変更などが原因で維持できないデータを変更する。変換ツールで大部分のデータを変更できますが、一部手動で変更しなければならない場合もあります。

ツールで変換される内容

- データベース構造の全体
- 大部分のデータ

ただしデータベースのテーブル、フィールドおよびリンクを参照するデータは確認が必要です。場合によっては手動変更する必要もあります。

この種のデータの一覧は、本マニュアルの「[参考情報](#)」の章、「[手動で変換するアプリケーションデータ](#)」の節にあります。

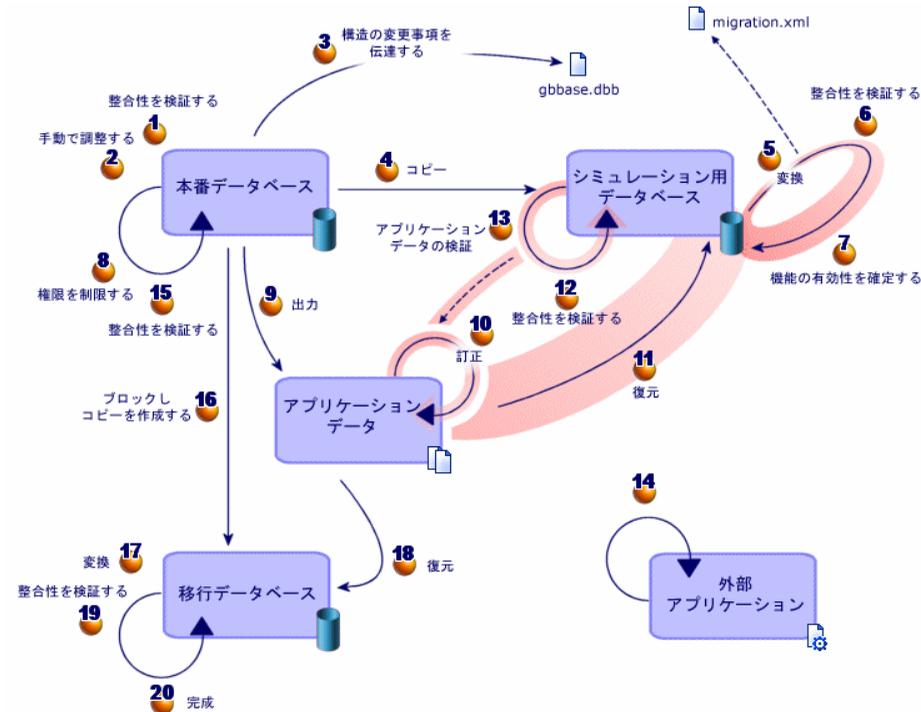
警告:

変換ツールをマイグレーションの目的以外に使用してはなりません。
ペレグリンシステムズ公認のエンジニアは、本マニュアルの指示に従って、任意管理項目を新規フィールドへ転送することもできます。

変換の実行方法

変換プロセスの概要は以下の通りです。

図 3.1. 変換 - プロセス



ツールを使用する場合もしない場合も、変換は以下の複数の段階を経て実行されます。

- 1 シミュレーション用データベース上で変換のシミュレーションを実行する。
 - 1 AssetCenter Database Administratorを使って旧フォーマットの本番データベースの整合性を検証します。
 - 2 AssetCenterを使って旧フォーマットの本番データベースを手動で調整します。
この段階では、旧フォーマットの本番データベースが変換可能になるように準備します。
 - 3 旧フォーマットの本番データベースに実行した構造の変更事項を、「標準 4.2.0 gbbase.dbb」データベース記述ファイルへ伝達します。
 - 4 旧フォーマットの本番データベースのコピーを作成します（このコピーは「シミュレーション用データベース」と呼ばれます）。シミュレーション用

- データベースで変換シミュレーションを試行している間、他のユーザは旧フォーマットの本番データベースを通常どおり使用することができます。
- 5 AssetCenter Database Administratorを使って、旧フォーマットのシミュレーション用データベースを変換します。必要に応じて、変換ファイル「migration.xml」を調整/テストします。
 - 6 AssetCenter Database Administratorを使って、4.2.0フォーマットのシミュレーション用データベースの整合性を検証します。
この操作では、変換がシミュレーション用データベースを破損していないかどうかを確認します。
 - 7 4.2.0フォーマットのシミュレーション用データベースの有効性を確定します。
この操作では、データが希望通りに変換されているかどうかを確認します。
 - 8 旧フォーマットの本番データベースへの権限の一部を制限し、ユーザがアプリケーションデータを変更できないようにします。
 - 9 AssetCenter Database Administratorを使って、手動で変換する必要があるアプリケーションデータを出力します。
 - 10 手動で変換するアプリケーションデータを、AssetCenter Script Analyzerを使って検証します。必要に応じてアプリケーションデータを訂正します。
 - 11 手動で変換したアプリケーションデータを、4.2.0フォーマットのシミュレーション用データベース内に復元します。AssetCenter Script AnalyzerまたはAssetCenter Database Administratorを使用します。
 - 12 4.2.0フォーマットのシミュレーション用データベースの整合性を、AssetCenter Database Administratorを使って検証します。
この操作では、復元がシミュレーション用データベースを破損していないかどうかを確認します。
 - 13 復元されたアプリケーションデータをAssetCenter 4.2.0を使ってテストします。
 - 14 AssetCenter 4.2.0と外部アプリケーションの統合の調整を準備します。
これにより、変換の最終段階にかかる時間を短縮できます。
- 2 旧フォーマットの本番データベースの2番目のコピー（「移行データベース」）を実際に変換する。
- 15 AssetCenter Database Administratorを使って、旧フォーマットの本番データベースの整合性を検証します。
 - 16 旧フォーマットの本番データベースをブロックし、コピー「移行データベース」を作成します。
 - 17 AssetCenter Database Administratorを使って、旧フォーマットの移行データベースを変換します。

18 手動で変換したアプリケーションデータを、4.2.0フォーマットの移行データベース内に復元します。これは、AssetCenter Script AnalyzerまたはAssetCenter Database Administratorを使って実行します。

19 AssetCenter Database Administratorを使って、4.2.0フォーマットの移行データベースの整合性を検証します。

20 変換を完了するために、AssetCenterを使って4.2.0フォーマットの移行データベースを完成させます。プログラムを更新した後に、この4.2.0フォーマットの移行データベースを、実際に使用します。

これらの手順の詳細については、本マニュアルの以下の箇所を参照にしてください。

- 「**段階を追ってマイグレーションを実行する - シミュレーション (シミュレーション用データベース)**」の章
 - **1** 「**手順1-旧フォーマットの本番データベースの整合性を検証する**」の節
 - **2** 「**手順2-旧フォーマットの本番データベースを手動調整する**」の節
 - **3** 「**手順3-旧フォーマットの本番データベースの構造の変更事項を伝達する**」の節
 - **4** 「**手順4-旧フォーマットの本番データベースのコピーを作成する**」の節
 - **5** 「**手順5-旧フォーマットのシミュレーション用データベースを変換する**」の節
 - **6** 「**4.2.0フォーマットのシミュレーション用データベースの整合性を検証する**」の節
 - **7** 「**4.2.0フォーマットのシミュレーション用データベースの有効性を確定する**」の節
 - **8** 「**手順8-旧フォーマットの本番データベースでの権限の一部を制限する**」の節
 - **9** 「**手順9-手動で変換するアプリケーションデータを出力する**」の節
 - **10** 「**手動で変換するアプリケーションデータを処理する**」の「**手順10-アプリケーションデータを検証し、修正する**」の節
 - **11** 「**手動で変換するアプリケーションデータを処理する**」の「**手順11-修正されたアプリケーションデータを復元する**」の節
 - **12** 「**手動で変換するアプリケーションデータを処理する**」の「**手順12-4.2.0フォーマットのシミュレーション用データベースの整合性を検証する**」の節
 - **13** 「**手動で変換するアプリケーションデータを処理する**」の「**手順13-復元されたアプリケーションデータを検証する**」の節
 - **14** 「**手順14-外部アプリケーションとの統合を調整する**」の節

- 「段階を追ってマイグレーションを実行する - 最終変換（移行データベース）」の章
 -  「手順15 - 旧フォーマットの本番データベースの整合性を検証する」の節
 -  「手順16 - 旧フォーマットの本番データベースをブロックしコピーを作成する」の節
 -  「手順17 - 旧フォーマットの移行データベースを変換する」の節
 -  「手順18 - 手動で変換されたアプリケーションデータを復元する」の節
 -  「手順19 - 4.2.0フォーマットの移行データベースで整合性を検証する」の節
 -  「手順20 - 4.2.0フォーマットの移行データベースを完成させる」の節

変換ツールの動作

変換ツールは以下のプログラムに統合されています。

- AssetCenter Database Administrator 4.2.0
- AssetCenter Script Analyzer 4.2.0

これらのプログラムは、AssetCenterのプログラムグループから起動します。

AssetCenter Database Administratorに統合された変換ツールには、以下のメニューからアクセスできます。

- **【アクション/データベースの診断/修復】**
このツールは、現在のデータベースの整合性を確認、修復します。
- **【マイグレーション/カスタマイズした構造の伝達】**
このツールは、旧フォーマットの本番データベースに実行された構造の変更情報を、「標準4.2.0 gbbase.dbb」データベース記述ファイルへ伝達します。
- **【マイグレーション/アプリケーションデータの出力】**
このツールは、手動変換するアプリケーションデータのコピーを、手動変換が可能なXMLフォーマットで出力します。
- **【マイグレーション/データベースの変換】**
このツールは、変換ファイル「migration.xml」の指示に従って、現在のデータベースの構造とデータを変換します。
- **【マイグレーション/アプリケーションデータの復元】**
このツールは、検証され修正されたアプリケーションデータをインポートします。

現バージョンと旧バージョンでの変換プロセスの相違点

旧フォーマットのデータベースの変換作業では、古いデータを空のデータベースにインポートするという、バージョン4.0.0より前の方法は使用しません。

変換ツールが、必要な変更を旧フォーマットの本番データベースで直接実行します。

この新しい方法には、以下の利点があります。

- 変換時間が大幅に短縮されます。
- 新しいデータベース構造にも存在するフィールドに保存されているデータは、変更されません。これらのデータをインポートする必要がないため、この分変更時間が短縮されます。
- 変換ファイル「migration.xml」のカスタマイズが可能です。
 - 変換ファイルの形式は、XML形式です。
 - 変換ファイルは、テキストエディタまたはXMLエディタで編集できます。
 - 変換ファイルは、多くの場合DBMSに依存しません。変換ファイルは、DBMSに固有のSQLステートメントに変換されます。

[マイグレーション/アプリケーションデータの出力]メニューで出力されたアプリケーションデータを再度復元する前に、AssetCenter Script Analyzerでデータを手動変換できます。

一部のアプリケーションデータを手動で変換する理由

データとパラメータの中には、自動的に変換できないものがあります。

Basicスクリプトを含むデータやパラメータがこれに当たります（これらは、AssetCenterのAQL言語を使用することもあります）。例えば、アクション、クエリ、フィールドのデフォルト値などです。

このようなデータの一覧に関しては、本マニュアルの「[参考情報](#)」の章、「[手動で変換するアプリケーションデータ](#)」の節を参照してください。

4.2.0バージョンがサポートしていないDBMSを使用するデータベースの、マイグレーション方法

旧フォーマットの本番データベースのDBMSが4.2.0バージョンにサポートされていない場合、以下の手順に従います。

- 1 旧フォーマットの本番データベースを、AssetCenter 4.2.0にサポートされているDBMSへ変換します。

変換方法については、マニュアル『[管理](#)』の「[AssetCenterデータベースの作成](#)」の章、「[DBMSを変更する](#)」の節を参照してください。

- 2 本マニュアルで説明されている手順に従って、マイグレーションを実行します。

調達モジュールに関する制約点

旧フォーマットの本番データベースを変換すると、以下の操作を実行できなくなります。

- データベース変換前に部分的に受領した注文品の残りを受領する
- 変換前に受領した品物を返却する

このため、旧フォーマットの本番データベースの変換前に上記の操作をあらかじめ実行しておくことをお勧めします。

マイグレーションの複雑な過程

本マニュアルで提案するマイグレーション方法は、多数の問題を予期し回避することを目的としています。

ただし、各ユーザ企業がAssetCenterを使用する方法に従って、本章の方法を調整する必要があります。

変換作業の複雑さは、旧フォーマットの本番データベースのカスタマイズの度合に応じて異なります。

4 | 段階を追ってマイグレーション を実行する - 準備段階（本番 データベース）

本章では、変換する前に必要な準備操作を段階を追って説明します。

事前分析

マイグレーションプロセスを実施する前に、以下の手順に従って各自のニーズと制約点を徹底的に分析する必要があります。

- 1 本マニュアルで説明されているマイグレーションの全段階を把握します。
- 2 AssetCenter 4.2.0に追加された変更事項を理解します。
詳しくは、本マニュアルの「[参考情報](#)」の章、「[その他のマニュアル](#)」の節を参照してください。
- 3 機能の変更事項がAssetCenterの使用に与える影響について予測します。
- 4 新しい機能を設置する時期を決定します（例：マイグレーション時、またはその後）。
- 5 上記の影響を考慮に入れて、プロジェクトの実施計画書を更新します（作業の割当、データの構成、パラメータの設定など）。
- 6 ユーザ用のマニュアルの更新と、ユーザ教育を計画します。

マイグレーションプロジェクトを開始する

バージョン4.2.0のAssetCenterは、旧バージョンに比べて大幅に改善されました。このため、以下のユーザをマイグレーションプロジェクトに含めるようお勧めします。

- データベースの体系的な分類や命名の担当者
- 以下の機能モジュール導入の担当者
 - 調達
 - 契約
 - ファイナンス
 - ケーブル
- 棚卸責任者
- データベースのカスタマイズの担当者
- レポート、クエリ、ワークフローチャート、アクションなどの作成担当者
- AssetCenterと外部アプリケーションの統合の担当者
- ユーザ教育の担当者
- ヘルプデスクの担当者

プロジェクトの開始時点から、上記の担当者を定義し、各自に関連情報を通知することが大切です。

ヒント:

旧バージョンの導入で使用した実施計画書を参照するようお勧めします。

プロジェクトの開始会議を開き、上記の担当者にマイグレーションの目的を説明した上で、タスクを分担し、スケジュールを決めます。

AssetCenterを高度な方法で使用する場合は（多数の整合性規則、自動処理機能、パラメータ設定がある場合など）、プロジェクトリーダーの傘下に機能分野または技術分野別のチームを設置します。

警告:

マイグレーションは技術的に複雑な作業です。このため各チームに有能なIT技術者を割り当てる必要があります。特に、デフォルトの変換ファイル「migration.xml」を変更する場合は、SQLの知識が不可欠になります。

新しい機能をすぐに利用する場合は、新しい実施計画書を作成し、パラメータを設定する必要があります。

 **ヒント:**

ペレグリンシステムズは、提携パートナー企業と共にマイグレーションをお手伝いします。経験豊富なコンサルタントが、プロジェクトのあらゆる段階でサービスを提供します。

ユーザおよびヘルプデスクエンジニアの教育

マイグレーションプログラムおよび旧フォーマットの本番データベースの変換と並行して、AssetCenterを使用するユーザの教育を計画する必要があります。

このためには、

- 1 教育の必須内容を定義します。
- 2 教育のスケジュールを立てます。
- 3 教材を準備します。
- 4 ユーザガイドを改訂します。

 **警告:**

AssetCenterに關与するユーザは、4.2.0フォーマットの本番データベースを実際に使用開始する前に、ユーザ教育を受けなければなりません。

 **ヒント:**

ペレグリンシステムズは、提携パートナー企業と共にマイグレーションをお手伝いします。経験豊富なコンサルタントがユーザ教育を担当します。

変換用コンピュータを準備する

旧フォーマットの本番データベースを変換するためには、変換用のコンピュータを準備しなければなりません。

本章では、変換用コンピュータに何をインストールするかについて説明します。

旧フォーマットの本番データベースに対応するバージョンのAssetCenterをインストールする

これは、以下の旧フォーマットのデータベースへアクセスするために必要になります。

- 本番データベース
- シミュレーション用データベース
- 移行データベース

少なくとも基本モジュールをインストールします。

旧フォーマットの本番データベースにアクセスできることを確認する

以下の操作を行うために、データベースへのアクセス権限が必要です。

- 変換時に旧フォーマットの本番データベースを準備する。
- 変換をシミュレーションし、実行するために、旧フォーマット本番データベースのコピーを作成する。

AssetCenter 4.2.0をインストールする

少なくとも以下のコンポーネントをインストールします。

- AssetCenterクライアント
- AssetCenter Database Administrator
- マニュアル
- ログビューア
- マイグレーション
- データキット
- AssetCenter Export

Connect-Itをインストールする（AssetCenter 4.2.0付属のバージョン）

手動変換されるアプリケーションデータを修正した後、このデータを復元する際に必要になります。

XMLファイルエディタをインストールする

テキストエディタを使用すれば十分ですが、変換ファイル「migration.xml」を編集し、ファイルのフォーマット（XML）が有効かどうかを確認するためには便利です。

Sun Java Runtime Environment（AssetCenter 4.2.0付属のバージョン）をインストールする

旧フォーマットの本番データベース構造のカスタマイズ情報を伝達するツールの使用時に、必要になります。

変換速度を左右する要素

- DBMSの性能
- AssetCenter Database Administratorのコンピュータと、旧フォーマットデータベースのコンピュータ間のデータ転送速度
- AssetCenter Database Administratorと、旧フォーマットデータベースがインストールされているコンピュータの性能（上記の要素ほど大切ではありません）

ヒント:

旧フォーマットの本番データベースのサイズが大きい場合、AssetCenter Database Administratorがインストールされているコンピュータと、旧フォーマットのデータベースをできる限り近づけなければなりません（例えばWANを経由しない、など）。特に長いフィールドやバイナリデータを含むテーブルでは注意が必要です（例：amComment、amImage）。

DBMSサーバを準備する

旧フォーマットのデータベースに十分な領域を割り当てます。

マイグレーション時には、旧フォーマットのシミュレーション用データベースと、旧フォーマットの移行データベースを変換する必要があります。

変換を適切に実行するために、各データベースに割り当てられる領域が十分であるかどうか確認してください。

必要な領域のサイズについては、本マニュアルの「サポートされる環境」の章、「DBMSサーバのディスク領域」の節を参照してください。

ロールバックセグメント (Rollback segments)

注意:

Rollback segmentsはOracle用語です。

Microsoft SQL ServerとSybase Adaptive Serverでは、これはトランザクションログ (transaction logs) に当たります。

全てのロールバックセグメントは、変換時に必要な最大トランザクションをサポートするように定義されなければなりません。

このトランザクションは、最大領域を占めるテーブル全体へ、1回の操作でINSERTを実行することです。

5 | 段階を追ってマイグレーション を実行する - シミュレーション (シミュレーション用データ ベース)

旧フォーマットの本番データベースを変換する前に、変換のシミュレーションを行う必要があります。

シミュレーションに本番データベースを使用することはできません。シミュレーション用データベースと呼ばれるコピーを使って、シミュレーションを実行します。

ユーザはこのため、シミュレーション中も旧フォーマットの本番データベースを通常どおり使用できるようになります。

シミュレーションを実行した後に、旧フォーマットの本番データベースの別のコピー（移行データベースと呼ばれます）を変換します。

4.2.0フォーマットの移行データベースが、実際に使用されます。

本章では、シミュレーション用データベースで実行する操作について、段階を追って説明します。

🔥 手順1 - 旧フォーマットの本番データベースの整合性を検証する

- 1  **重要項目:**
旧フォーマットの本番データベースのバックアップコピーを作成します。
- 2 まず第一に旧バージョンのAssetCenter Database Administratorを使って、整合性を検証します。
 - 1 旧バージョンのAssetCenter Database Administratorを起動します。
 - 2 旧フォーマットの本番データベースに接続します（[ファイル/開く/既存のデータベースを開く]メニュー）。
 - 3 データベースの診断画面を表示します（[アクション/データベースの診断/修復]メニュー）。
 - 4 テーブルのリストで[(テーブルなし)]を選択します（実際には[(すべてのテーブル)]を選択したことを意味します）。
 - 5 ログファイルの名前とパスを指定します。
 - 6 [レコードの整合性のチェック]オプションを選択します。
 - 7 [修復]オプションを選択します。
 - 8 [実行]をクリックします。
 - 9 実行画面のメッセージを確認します。
 - 10 必要に応じて、ログファイルの内容を確認します。

ヒント:

ログビューアを使用すると、ログファイルを閲覧できます。

- 3  **警告:**
旧フォーマットの本番データベースのDBMSがDB2である場合、以下の検証作業を行う必要はありません。
次に、バージョン4.2.0のAssetCenter Database Administratorを使って、整合性を検証します。
 - 1 AssetCenter Database Administrator 4.2.0を起動します。
 - 2 旧フォーマットの本番データベースに接続します（[ファイル/開く/既存のデータベースを開く]メニュー）。
 - 3 データベースの診断画面を表示します（[アクション/データベースの診断/修復]メニュー）。
 - 4 テーブルのリストで[(すべてのテーブル)]を選択します。

- 5 ログファイルの名前とパスを指定します。
- 6 **【レコードの整合性のチェック】** オプションを除くすべての検証オプションを選択します。
- 7 **【修復】** オプションを選択します。
- 8 **【実行】** をクリックします。
- 9 実行画面のメッセージを確認します。
- 10 必要に応じて、ログファイルの内容を確認します。

 **ヒント:**

ログビューアを使用すると、ログファイルを閲覧できます。

分析 / 修復プログラムの詳細については、マニュアル『**管理**』の「**データベースの診断と修復**」の章を参照してください。

手順2 - 旧フォーマットの本番データベースを手動調整する

調整作業（旧フォーマットの本番データベースの全バージョン用）

【amCounter】テーブルを更新する

本節の内容は、【amCounter】テーブルを管理するup_GetCounterValストアードプロシージャを、以下のテクノートの指示に従って変更したユーザを対象としています。

- Microsoft SQL Server : TN317171736
- Sybase Adaptive Server : TN941931
- Oracle Workgroup Server : TN12516652
- DB2 UDB : TN1029175140

上記のテクノートの指示通りに変更を実行した場合、up_GetCounterValストアードプロシージャは、【amCounter】テーブルの一部のレコードを更新できなくなります。

このため、**旧フォーマットの本番データベース**を変換する前に以下の操作を実行します。

- 1 ストアドプロシージャup_GetCounterValを変換後に同じ方法で変更する場合は、コピーを作成します。

- 2 【amCounter】テーブルから別のテーブルへ派生されたカウンタを手動で変更します。
- 3 up_GetCounterValストアプロシージャを、初期状態に戻します。

 **ヒント:**

手順「[手順20 - 4.2.0フォーマットの移行データベースを完成させる](#)」で、テクノートの指示を再適用します。

調整作業（旧フォーマットの本番データベースが4.0.0バージョンより前の場合）

 **警告:**

本節の内容は、4.0.0またはそれ以降のバージョンの旧フォーマットの本番データベースを変換するユーザには当てはまりません。

上記のユーザは、旧フォーマットの本番データベースに手動で操作を加える必要はありません。

 **警告:**

本節で説明されている手動調節を実行する前に、旧フォーマットの本番データベースのバックアップコピーを作成しておくことをお勧めします。

旧フォーマットの本番データベースの変換を正しく進めるためには、変換前に特定のデータを変更する必要があります。

旧フォーマットの本番データベースの制約事項の大部分は、「migration.xml」変換ファイルのMapping要素により導入 / 作成されます。

本節では、標準の変換ファイルが導入する制約事項について説明します。標準の変換ファイルを自分で変更する場合は、変更事項がもたらす制約点を確認し、検証するようにしてください。

必須フィールドと必須リンク

あるテーブル内にレコードを作成するためには、必須フィールドと必須リンクに値が入力されていなければなりません。

フィールドやリンクが必須であるかどうかは、AssetCenterデータベース内、または「gbbase.dbb」データベース記述ファイル内で定義されます。

この必須性は、全てに当てはまる場合もあれば、スクリプトの計算によって適用される場合もあります。

変換プログラムが作成または変更したレコードは、「カスタマイズ済4.2.0 gbbase.dbb」データベース記述ファイル内に明記されている、フィールドやリンクの必須属性に従わなければなりません。

必須フィールドやリンクは、明示的な方法（「migration.xml」変換ファイルで説明されている）または暗示的な方法（フィールドやリンクが同じSQL名であるために、自動的に推測される）で、関連付けられていなければなりません。

AssetCenter 4.2.0と共にデフォルトでインストールされる「migration.xml」変換ファイルは、**旧フォーマットの本番データベース**のフォーマットと、「標準4.2.0 gbbase.dbb」データベース記述ファイルのフォーマットが変更されていない場合、適切に機能するようになっています。

標準の変換ファイル「migration.xml」は、以下の場合には適切でない場合もあります。

- **旧フォーマットの本番データベース**の使用中に、フィールドやリンクの必須属性を削除した場合
- 「標準4.2.0 gbbase.dbb」データベース記述ファイルの一部のフィールドとリンクに、必須属性を追加した場合

必須フィールドとリンクに値を入力するために、**旧フォーマットの本番データベース**の一部のデータを、変換プログラムが使用する場合があります。

「カスタマイズ済4.2.0 gbbase.dbb」データベース記述ファイル内で必須と宣言されているフィールドやリンクが、**旧フォーマットの本番データベース**で入力されているかどうか、変換前に確認してください。

例えば、[amAsset] テーブルの [lCategId] フィールドは、**旧フォーマットの本番データベース**内で記入されていなければなりません。

不明な場合は、この外部キーに値が入力されているかどうか確認してください。

フィールド値の長さ

旧フォーマットの本番データベースのフィールドの一部は、4.2.0 **フォーマットの本番データベース**の別のフィールドに値を入力するために使用されます。

これらのソースフィールドの一部は、ターゲットフィールドよりも長い場合があります。

変換時に問題が起こった場合は、ソースフィールドに格納されている値の長さが、ターゲットフィールドのサイズを超過しないかどうか確認する必要があります。

値がフィールドのサイズを超過している場合、以下の方法で問題を解決します。

- ソース値の長さを短縮します。
- ターゲットフィールドのサイズを拡大します（「カスタマイズ済4.2.0 gbbase.dbb」ファイル内で）。

長すぎる値は変換時に切り捨てられます。

文字「^」

この文字を、旧フォーマットの本番データベースのフィールドの値、特に以下のフィールドの値に使用してはなりません。旧フォーマットの本番データベースのバージョンに、これらのフィールドが含まれているかどうか確認してください。

表 5.1. 文字「^」を含んではならないフィールド - 一覧

テーブルのSQL名	フィールドのSQL名
amProduct	Model
amProduct	CatalogRef
amSoftware	Name
amCatalog	Code
amCompany	Code
amCompany	Name
amProdSupp	PriceCur
amCatProduct	FullName
amAccessRestr	SQLName
amAssetRent	Code
amBrand	BarCode
amBudgClass	Code
amBudgClass	Name
amBudget	Code
amBudget	Name
amBudget	Type
amBudgetCategory	Code
amCategory	Name
amCategory	BarCode
amCategory	FullName
amCategory	sLvl
amCntrRent	Code
amDateAlarm	Code
amDeprScheme	Code
amEscSchLevel	Code
amFloorPlan	Code
amFuncDomain	SQLName
amFuncDomain	Name
amReservation	ItemNo
amLocation	BarCode
amLocation	FullName

テーブルのSQL名	フィールドのSQL名
amLocation	Name
amLossValRule	Code
amModel	BarCode
amModel	FullName
amModel	Name
amContract	Ref
amNature	Code
amNature	Name
amNews	Topic
amPeriod	Name
amPeriod	Code
amEstimate	PONumber
amEstimate	EstimNumber
amPOrdLine	FullName
amPOrdLine	ItemNo
amEstimLine	FullName
amEstimLine	ItemNo
amPortfolio	Code
amPortfolio	FullName
amConsUse	ItemNo
amAsset	FullName
amAsset	AssetTag
amProdCompo	FullName
amProfile	SQLName
amProject	Code
amReceipt	ReceiptNumber
amRequest	ReqNumber
amSoftLicCounter	Code
amThirdParty	Code
amUserRight	SQLName
amPOrder	PONumber

製品明細

リンクの入れ子構造が以下の通りである場合、

製品P1 -> 製品P1の明細C1 -> 明細C1に対応する製品P2 -> 製品P2の明細C2 -> 明細C2に対応する製品P3

- 「製品P1 -> 製品P1の明細C1 -> 明細C1に対応する製品P2」全体は変換されます。

- 「製品P2 -> 製品P2の明細C2 -> 明細C2に対応する製品P3」全体は変換され
ます。
- これに対して、リンクの入れ子構造は、P2とC2のレベルで中断されます。
これは、P3がP1を構成するという情報が消失することを意味します。

P3とP1間のリンク情報を保存する場合、製品P1へ新規明細C3を追加し、さらにP3をC3に関連付ける必要があります。

この操作は変更前に実行する必要があります。

ライセンス契約

ライセンス契約は、「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」の章、「[手順5 - 旧フォーマットのシミュレーション用データベースを変換する](#)」の節の「[ライセンス契約](#)」で説明されている手順に従って変換してください。

上記の節で説明されている方法でライセンス契約を変換せずに、契約が契約のままであるように処理するには、

- 1 【amContract】テーブル内に残しておくライセンス契約に関連付けられた全資産の、【ILicCtrId】フィールドの値を、「0」にします。
- 2 必要に応じて、【AstCntrDescs】リンクを使ってこれらの資産を上記の契約にさらに関連付けます（この操作は、中間テーブル【amAstCntrDesc】にレコードを作成します）。

基本調整

【IAdjustId】外部キーが、【amFieldAdjust】テーブルの全レコード用に、「0」以外の値であることを確認してください。

リストデータの値

【Value】フィールドが、【amItemListVal】テーブルの全レコード用に、NULLではないことを確認してください。

調達モジュールとワークフローモジュール

データベース変更前に、進行中のプロセス（部分受領した発注、返却する資産、進行中のワークフローなど）の数をできる限り最小限に抑えてください。

特に、ライセンス契約を使用するワークフローのインスタンスは、変換中に削除されます。

 **注意:**

ライセンス契約は、【amContract】テーブルのレコードです。

- これらのレコードでは、【seType】フィールドの値は「5」です。
- レコードは、外部キー【ILicCntrId】（【amAsset】テーブル）により、少なくとも1つの資産に関連付けられています。

 **警告:**

変換後に問題が起こった場合に参照できるように、旧フォーマットの本番データベースのコピーを厳重に保存しておいてください。

製品のサプライヤ

【amProdSupp】テーブルは、4.0.0バージョンからなくなります。

変換時に、【amProdSupp】テーブルの【mPrice】フィールドの通貨が、【amCurrency】テーブル内で以下の方法で宣言されていると、【amProdSupp】テーブルのレコードは、【amCatRef】テーブルへ転送されます。

- デフォルトの通貨
- 基本通貨1
- 基本通貨2

この条件を満たさない【amProdSupp】内のレコードは変更されません。

他の通貨を管理する必要がある場合は、以下の方法を使用します。

- 旧フォーマットの本番データベースの変換前に、【mPrice】フィールドを適切な通貨に変換する。

 **ヒント:**

ユーロゾーンの通貨をユーロへ換算するためのユーティリティを入手するには、ペレグリンシステムズのテクニカルサポート迄お問合せください。

- 他の通貨を以下の要素に再び割り当てる。

- デフォルトの通貨
- 基本通貨1
- 基本通貨2

現在割り当てられている通貨が、旧フォーマットの本番データベースで使用されていない場合は、

- 処理する追加通貨ごとに、Mapping要素を「migration.xml」ファイルへ追加する。

詳しくは、本マニュアルの「[変換ファイル「migration.xml」の調整](#)」の章を参照してください。

Mapping要素の例が、「migration.xml」変換ファイル内に提案されています。

この要素を見つけるには、変換ファイルを開き、「Use the following mapping to add another currency」というテキストを検索します。

デフォルトの「migration.xml」ファイルを使って、変換ツールは【amCatalog】テーブル内で、サプライヤごとにレコードを3つまで作成します（サポートされる各通貨ごとに1レコード）。

【amCatRef】テーブルの複数のリファレンスは、変換時にこれらのカタログの1つに関連付けられます。

見積

変換時に、【amEstimate】テーブルのレコードは【amPOOrder】テーブルへ転送されます。【seStatus】フィールドの値は「見積済」になります。

【IPOrdLineId】フィールドの値が「0」以外の見積明細を含む見積は、変換時に削除されます（これは、【見積】テーブルのレコードは発注のレコードに転換されたと思われるためです。そしてこの発注のレコードが変換されます。これはAssetCenterバージョン4.2.0で、見積を管理する方法です）。

【amPOOrder】テーブルに負荷をかけないために、【amEstimate】テーブルの不必要な見積を、変換前に削除しておくこともできます。

見積を保存する場合は、変換時に保存しておく全見積明細の【IPOrdLineId】フィールドに、「0」を入力する必要があります。

製品明細

変換が適切に機能するのは、製品明細（【amProdCompo】テーブル）のツリー構造のレベル数が9以下の場合です（0から8）。

この条件を満たすために、【sLvl】フィールドが、ツリー構造で9以上である製品の構造を変更します。

ヒント:

より多くのレベルを管理できるように、変換ファイルのスキプトを変更することもできます。

- 1 使用中のDBMSに対応する<PostActions>要素内で、以下の行のペアを検索します。

```
UPDATE amCatProduct SET FullName = Q.FullName || amCatProduct.InternalRef
|| '/', sLvl = Q.sLvl + 1 FROM amCatProduct, amCatProduct Q WHERE amCatProduct.sLvl = -1 AND Q.ICatProductId = amCatProduct.IParentId AND Q.sLvl <> -1
GO
```

これらの行のペアの数は、サポートされているレベルの数に当たります。

- 2 サポートする追加階層レベルごとに、行のペア1行を追加します。

- 【amProdCompo】テーブルのレコードが以下の要素両方にリンクしている場合、
- フィールド【bSuppPackage】が「1」である主製品（MainProductリンク）
 - 【UsedAsset】リンクによる資産、または【UsedContract】リンクによる契約
- この場合、【UsedAsset】リンクまたは【UsedContract】リンクは、変換時に転送されません。
- これらのリンクを転送する場合は、主製品の【bSuppPackage】フィールドの値を「0」に変更します。

発注明細

変換が適切に機能するのは、発注明細（【amPOrdLine】テーブル）のツリー構造のレベル数が10以下の場合です。

この条件を満たすために、【sLvl】フィールドが10以上である発注明細を、ツリー構造内で上位へ移動させます。

ヒント:

より多くのレベルを管理できるように、変換ファイルのスクリプトを変更することもできます。

- 1 使用中のDBMSに対応する<PostActions>要素内で、以下の行のペアを検索します。

```
UPDATE amPOrdLine SET FullName = Q.FullName || amPOrdLine.ItemNo || '/', s
Lvl = Q.sLvl + 1 FROM amPOrdLine, amPOrdLine Q WHERE amPOrdLine.sLvl
= -1 AND Q.IPOrdLinId = amPOrdLine.IParentId AND Q.sLvl <> -1
GO
```

これらの行のペアの数は、サポートされているレベルの数に当たります。

- 2 サポートする追加階層レベルごとに、行のペア1行を追加します。

カテゴリ

変換が適切に機能するのは、カテゴリ（amCategoryテーブル）のツリー構造のレベル数が10以下の場合です。

この条件を満たすために、【sLvl】フィールドが10以上であるカテゴリを、ツリー構造内で上位に移動させます。

予算

デフォルトの「migration.xml」変換ファイルにより、【amBudget】テーブルの内容は、【amCostCategory】テーブルへ転送されます。

「予算」を分類（経費を属性ごとにまとめる）目的で使用していた場合、この動作は適切ですが、予算額などを管理していた場合には不適切です。

予算額を管理するために「予算」を使用していた場合は、「migration.xml」変換ファイルを調整し、予算が【amBudgLine】テーブルへ転送されるように設定しなければなりません。

予算を【amBudgLine】テーブルへ転送するためのMapping要素は、有効にされないまま「migration.xml」ファイル内に挿入されています。これらの要素を使うと、【amBudget】と【amBudgLine】を関連付けられるようになります。

Mapping要素を変換時にアクティブにすると、

- 【dStart】フィールドと【dEnd】フィールドが入力されているか否かにより、予算（amBudget）が処理される方法が変化します。
 - 2つのフィールドの内1つが入力されていない場合、変換プログラムは、【コストカテゴリ】（SQL名：amCostCategory）テーブル内のレコードを移動させます。
 - 2つのフィールドが入力されている場合、変換プログラムは、【予算明細】（SQL名：amBudgLine）テーブルのレコードと、【コストカテゴリ】（SQL名：amCostCategory）テーブルのレコードを移動させます。
- このため、変換時に移動させるレコードの種類に応じて、【dStart】と【dEnd】フィールドを入力する必要があります。

【完全名】フィールド

【完全名】（SQL名：FullName）フィールドに入力される文字列に、文字「/」が含まれている場合、この文字は階層レベルの区切り文字として解釈されます。一部のDBMSでは、文字「/」を別の文字と置換するように変換ファイルで設定できるので、問題は生じません。

Sybase Adaptive Serverを使用している場合、仮の置換を実行することは不可能です。

この場合、【完全名】フィールドへのデータ入力に使用されるフィールドの文字「/」を、別の文字に置き換える必要があります。

確認が必要なフィールドは以下の通りです。

表 5.2. 文字「/」を含んではならないフィールド - 一覧

テーブルのSQL名	フィールドのSQL名	関連するバージョン						
		3.0.1	3.0.2	3.1.0	3.5.0	3.5.1	3.6.0	4.0.0
amItemListVal	Value (メーカー用)	はい	はい	はい	はい	はい	はい	はい
amFamily	Brand	はい	はい	はい	はい	はい	はい	はい
amFamily	Name	はい	はい	はい	はい	はい	はい	はい
amAsset	ComputerName							はい
amAsset	AssetTag	はい	はい	はい	はい	はい	はい	はい
amInvoice	InvoiceNumber	はい	はい	はい	はい	はい	はい	はい

テーブルのSQL名	フィールドのSQL名	関連するバージョン						
		3.0.1	3.0.2	3.1.0	3.5.0	3.5.1	3.6.0	4.0.0
amProduct	Model	はい	はい	はい	はい	はい	はい	
amSoftware	Publisher	はい	はい	はい	はい	はい	はい	
amSoftware	Name	はい	はい	はい	はい	はい	はい	
amSoftware	VersionLevel	はい	はい	はい	はい	はい	はい	
amContract	Ref	はい	はい	はい	はい	はい	はい	
amAdjustment	ItemNo	はい	はい	はい	はい	はい	はい	
amConsUse	ItemNo	はい	はい	はい	はい	はい	はい	
amComputer	Name							はい

基本調整

【基本調整】（SQL名：amFieldAdjust）テーブルのレコード用に【調整】（SQL名：Adjustment）リンクが入力されていないと、【基本調整】テーブルのレコードは消失します。

このため、変換するレコードがこの条件を満たしているかどうかを、変換前に確認する必要があります。

機能ドメイン

【amFuncDomain】テーブルのレコードの【名前】（SQL名：Name）フィールドは、SQL名の規則に従わなければなりません（英数字と文字「_」のみが許可されています）。これは、4.2.0データベース内の【SQL名】（SQL名：SQLName）フィールドにデータを入力するために、これらのレコードが使用されるためです。

3 手順3 - 旧フォーマットの本番データベースの構造の変更事項を伝達する

警告:

この操作を実行するには、以下で使用される「標準4.2.0gbbase.dbb」データベース記述ファイルが、AssetCenter 4.2.0と共にインストールされる、未変更の標準ファイルでなければなりません。

この操作の特徴は以下の通りです。

- 旧フォーマットの本番データベースの標準構造を変更（テーブル、フィールド、インデックスの追加や変更）したユーザが、4.2.0フォーマットの本番

データベースでもこれらの変更事項を保持する場合に、この操作が必要になります。

- この操作の目的は、「標準4.2.0 gbbase.dbb」ファイル内に、構造の変更事項を伝達することです。

 **ヒント:**

このようにして作成された「標準4.2.0 gbbase.dbb」ファイルは、変換時に4.2.0 **フォーマットのデータベース**の構造を作成するために使用されます。

- この操作を実行するには、AssetCenter Database Administratorからアクセスできる専用ツールを使用します。

 **警告:**

AssetCenter Database Administratorを用いて**旧フォーマットの本番データベース**に実行された構造の変更事項のみが、有効になります。

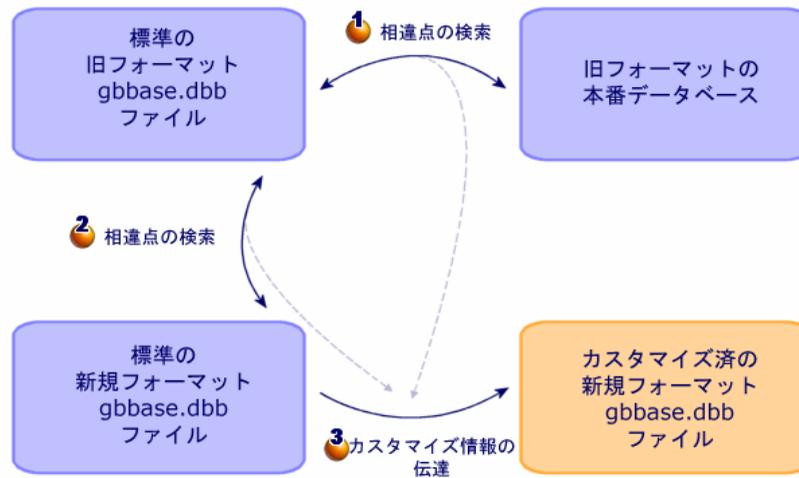
他の方法で構造を変更した場合、全構造変更を手動で取り消す必要があります。

伝達される構造パラメータの一覧に関しては、本マニュアルの「[参考情報](#)」の章、「[旧フォーマットの本番データベースの伝達された構造パラメータ](#)」の節を参照してください。

一般的な機能

構造の変更事項を伝達する仕組みは以下の図の通りです。

図 5.1. 構造の変更事項の伝達 - プロセス



手順 **1** (🔴) : ツールは、旧フォーマットの本番データベースの構造を、「標準の旧フォーマット gbbase.dbb」ファイルと比較し、相違点を検索します。

手順 **2** (🟠) : ツールは、「標準の旧フォーマット gbbase.dbb」ファイルと「標準4.2.0 gbbase.dbb」ファイルと比較し、相違点を検索します。

手順 **3** (🟡) : ツールは、「標準4.2.0 gbbase.dbb」ファイルを、手順 **1** と **2** で検索された相違点に応じて、コピー/変更します。コピー/変更時には以下の規則に従います。

- バージョン4.2.0には存在しなくなった標準テーブルに追加された変更事項は、消失します。
- 手順 **1** と **3** で、同じテーブル、フィールドまたはリンクに関する変更事項が検索された場合、手順 **2** で検出された変更事項が優先されます。この場合警告メッセージが表示されます。

🔪 注意:

例外 : 手順 **1** と **2** で、同じテーブル、フィールドまたはリンク内で **[タイトル]** や **[説明]** の変更が検出されると、手順 **1** で検出された変更事項が優先されます。

- 構造の変更を最終的に伝達する前に、バージョン4.2.0内に同じSQL名で表示されるテーブル、フィールドとインデックスのSQL名を、旧フォーマットの本番データベース内で変更する必要があります。

変更しないと、同名の4.2.0バージョンの標準フィールドとの競合が発生します。

「カスタマイズ済4.2.0 gbbase.dbb」ファイルは、明確に識別されなければなりません。このファイルは次の段階で使用されます。

- 手順9 - 手動で変換するアプリケーションデータを出力する
- 手動で変換するアプリケーションデータを処理する
- 手順5 - 旧フォーマットのシミュレーション用データベースを変換する
- 手順17 - 旧フォーマットの移行データベースを変換する

構造の変更事項の伝達

- 1 AssetCenter Database Administrator 4.2.0を起動します。
- 2 旧フォーマットの本番データベースに、「Admin」ログインで接続します（[ファイル/開く/既存のデータベースを開く]メニュー）。
- 3 [アクション/カスタマイズした構造の伝達]メニューを選択します。
- 4 [生成フォルダ]に値を入力します。この空のフォルダに、「カスタマイズ済4.2.0 gbbase.dbb」ファイルが作成されます。
- 5 [ビルド]をクリックします。
- 6 画面に表示されるメッセージを読みます。
ログファイル「newddb.log」を参照します（ファイルは、[生成フォルダ]フィールドで指定されるフォルダにあります）。

ヒント:

ログビューアを使用すると、ログファイルを閲覧できます。

- 7 メッセージの指示に従って、旧フォーマットの本番データベースの構造を変更し、手順「手順4 - 旧フォーマットの本番データベースのコピーを作成する」の段階から、マイグレーション操作を再開します。
「カスタマイズ済4.2.0 gbbase.dbb」ファイルが、エラーメッセージなしの正確なファイルになるまで、この手順を繰り返します。

8 警告:

AssetCenter 4.1.0またはそれ以降のバージョンを移行するユーザは、この手順を実行する必要がありません。

一部のスクリプトを、「標準4.2.0 gbbase.dbb」ファイルへ伝達できない場合もあります。

スクリプトが伝達されないと、各スクリプトごとにメッセージがログファイル「newdbb.log」に作成されます。また「.xml」ファイルが、「<生成フォルダ>/dbbscripts」フォルダと「<生成フォルダ>/builddb/dbbscripts」フォルダ内に作成されます。

これらのカスタマイズ情報は、「カスタマイズ済4.2.0 gbbase.dbb」ファイル内に、手動で伝達しなければなりません。

スクリプトをAssetCenter Script Analyzerを使って変換する場合は、「**手順10 - アプリケーションデータを検証し、修正する**」の段階でこの操作を実行できます。

AssetCenter Script Analyzerは変更方法の候補を表示します。AssetCenter Database Administratorを使って、「カスタマイズ済4.2.0 gbbase.dbb」ファイル内で手動変更します。

- バージョン4.0.0以降の旧フォーマットの本番データベースを変換する場合は、追加した各ページが有効であるかどうかを、AssetCenter Database Administratorで確認します。有効でない場合は手動で訂正してください。

警告:

「**手順5 - 旧フォーマットのシミュレーション用データベースを変換する**」手順の実行中に、「カスタマイズ済4.2.0 gbbase.dbb」ファイルを新たに変更します。

競合の可能性

構造の変更事項の伝達が不意に中断した場合は、Javaのインストール先フォルダの「/jre/lib/ext」サブフォルダに、「xerces.jar」ファイルがあるかどうか確認してください。

ファイルが上記のサブフォルダ内にある場合、このファイルの場所を一時的に変更し、構造の変更事項の伝達を再び実行します。

構造の変更事項を変換ファイル「migration.xml」に適用する

警告:

AssetCenter 4.1.0またはそれ以降のバージョンを移行するユーザは、以下の手順を実行する必要がありません。

伝達された構造の変更事項に「テーブルの追加」タイプの変更が含まれている場合は、テーブルの変換を管理できるようにするために、「migration.xml」変換ファイルを変更する必要があります。

🔥 手順4 - 旧フォーマットの本番データベースのコピーを作成する

💡 従来のコピーの問題点

DBMSのツールを使って旧フォーマットの本番データベースをコピーする場合、AssetCenter Database Administrator以外のツールで実行された以下の要素の追加、変更または削除もコピーされるため、旧フォーマットの本番データベースのコピーは元のデータベースと同一になります。

- インデックス
- トリガ
- ストアドプロシージャ
- ビュー

しかし変換プログラムは、この種の構造変更を管理することはできません。

このため旧フォーマットの本番データベースの変換前に、構造の変更事項を取り消す必要があります。

変換に適したコピーを作成するには、以下の2つの方法があります。

- DBMSツールを使ってコピーを作成し、上記の要素の構造変更を取り消します。
- AssetCenter Database Administratorを使って、旧フォーマットの本番データベースのコピーを空のデータベース内に作成します。

🔪 注意:

変換用コンピュータから、旧フォーマットの本番データベースのコピーにアクセスできなければなりません。

データベースのコピーの作成方法については、DBMSの付属マニュアルを参照してください。

🔧 方法1 : DBMSツールで旧フォーマットの本番データベースをコピーする

- 1 DBMSツールで旧フォーマットの本番データベースをコピーします。
作成されたコピーは、元の旧フォーマットの本番データベースと全く同一です。
- 2 以下の要素に実行された全変更事項を取り消します。
 - インデックス
 - トリガ
 - ストアドプロシージャ

- ビュー
- 3 旧フォーマットのシミュレーション用データベースへ、AssetCenter接続を作成します。

方法2 : AssetCenter Database Administratorを使って旧フォーマットの本番データベースのコピーを作成する

- 1 旧フォーマットの空のAssetCenterデータベースを作成します。
- 2 この空のデータベースへAssetCenter接続を作成します。
- 3 AssetCenter Database Administratorで旧フォーマットの本番データベースを開きます。
- 4 旧フォーマットの本番データベースのコピーを、作成しておいた空のデータベース内に作成します（[アクション/データベースを空のデータベースにコピー]メニュー）。

この方法は、上記の要素に実行された全変更事項を取り消す方法よりも優れています。

AssetCenter Database Administratorで、旧フォーマットの本番データベースのコピーを空のデータベース内に作成する方法については、マニュアル『管理』の「テスト用データベースの使用」の章、「本番データベースのコピーを作成する」の節を参照してください。

手順5 - 旧フォーマットのシミュレーション用データベースを変換する

警告:

変換ツールを、4.2.0フォーマットの本番データベースの構造を変更する（テーブル、フィールド、インデックス、ストアドプロシージャ、トリガ、ビューなどの追加、削除、変更）ために使用してはなりません。

この種の変更事項はマイグレーション後に計画してください。

🔑 変換ファイル「migration.xml」の調整

⚠️ 警告:

この操作は、ペレグリンシステムズ公認のマイグレーション用エンジニアが実行しなければなりません。

公認エンジニア以外のユーザが操作を実行した場合、ペレグリンシステムズは一切責任を負いません。

AssetCenter 4.2.0は、デフォルトの変換ファイルと共にインストールされます（マイグレーションでサポート可能な旧バージョンのAssetCenterにつき1ファイル）。変換ファイルは、**旧フォーマットのシミュレーション用データベースの変換時にどのデータをどのように変換するのかを説明しています。**

変換ファイルの名前は、「migration.xml」です。

変換ファイルは一般的に、「C:\Program Files\Peregrine\AssetCenter\migration\fromxxx」フォルダに格納されています。「xxx」が旧バージョンの番号に当たります。

一般的な方法でAssetCenterを使用していた場合、デフォルトでインストールされている変換ファイルの内1つが使用可能なはずですが。

各ユーザ企業の必要に応じて（例：任意管理項目を新しいデータベース構造のフィールドに転送する、デフォルトの用途以外でフィールドを使用する、テーブルとフィールドを追加するなど）、変換ファイルを調整しなければなりません。

⚠️ 警告:

標準またはカスタム変換ファイルを、次の段階で**移行データベース**上で実行する前に、必ず**シミュレーション用データベース**でテストしてください。

変換ファイルのシンタックスおよびカスタマイズ方法に関しては、本マニュアルの「**参考情報**」の章、「**変換ファイル「migration.xml」の調整**」の節を参照してください。

変換ファイルの変更が、**旧フォーマットの本番データベース**のデータにもたらす制約点については、「**段階を追ってマイグレーションを実行する-シミュレーション（シミュレーション用データベース）**」の章、「**手順2-旧フォーマットの本番データベースを手動調整する**」の節を参照してください。

 **重要項目:**

「migration.xml」変換ファイルをカスタマイズする前に、ファイルの名前や場所を変更してはなりません。これは、このファイルを使用するツールが、ファイルを標準フォルダ内で検索するためです。

このため、変換ファイルを変更する前に、バックアップコピーを作成しておくことをお勧めします。

旧フォーマットのシミュレーション用データベースを変換する

旧フォーマットのシミュレーション用データベースを変換するには、以下の手順に従います。

- 1 **本番データベース**のDBMSがDB2である場合、「db2udb72.sql」ファイルにあるSQL命令を**旧フォーマットのシミュレーション用データベース**に対して実行します。一般的に「db2udb72.sql」ファイルへのパスは、「C:\Program Files\Peregrine\AssetCenter\migration\db2udb72」です。
- 2 AssetCenter Database Administratorを起動します。
- 3 **旧フォーマットのシミュレーション用データベース**に、Adminログインで接続します（**[ファイル/開く/既存のデータベースを開く]**メニュー）。

 **重要項目:**

AssetCenterの接続の詳細画面では、以下の点に注意します。

- **[所有者]**フィールドに値を入力してはなりません。
- **[ユーザ]**フィールドは、データベーステーブルの所有者であるユーザ（データベースの全種のオブジェクトの作成権限があるユーザ）を参照しなければなりません。
- Microsoft SQL Serverでは、テーブルの所有者がdboである場合、接続ログインは、dbo.<テーブル>の形でデフォルトのテーブルを作成しなければなりません（特にログインsaの場合）。

- 4 **[マイグレーション/データベースの変換]**を選択します。
- 5 **[ターゲットデータベースの記述ファイル]**フィールドに、「カスタマイズ済4.2.0gbbase.dbb」ファイルの完全パスを入力します（このファイルは、**[マイグレーション/カスタマイズした構造の伝達]**メニューの**[生成フォルダ]**フィールドで選択したフォルダ内にあります）。

「カスタマイズ済4.2.0 gbbase.dbb」ファイルの作成については、本マニュアルの**「段階を追ってマイグレーションを実行する - シミュレーション（シ**

ミュレーション用データベース)」の章の、「手順3 - 旧フォーマットの本番データベースの構造の変更事項を伝達する」の節を参照してください。

- 6 **【ライセンスファイル】**フィールドに、AssetCenter 4.2.0のライセンスファイル「license.cfg」の完全パスを入力します（このファイルはAssetCenter 4.2.0に付属しています）。

 **注意:**

AssetCenter 4.0.0以降では、ライセンスファイルがデータベースへのアクセスを管理します。

ライセンスファイルの詳細については、マニュアル『管理』の「AssetCenter データベースの作成」の章、「ライセンスファイルを選択する」を参照してください。

- 7  **警告:**

AssetCenter 4.1.0またはそれ以降のバージョンを移行するユーザは、この手順を実行する必要がありません。

【変換ファイル】フィールドで、旧フォーマットのシミュレーション用データベースのバージョンに対応する変換ファイルを指定します（一般的にパスは、C:\Program Files\Peregrine\AssetCenter\migration\fromxxx\migration.xmlです）。

- 8 **【ログフォルダ】**フィールドで、変換ログファイル「sdu.log」の保存用フォルダを指定します（例：C:\temp\）。
- 9 **【更新】**をクリックします。
- 10 画面に表示されるメッセージを読みます。

 **ヒント:**

パラメータ**【入力タイプ】**が**【コメント】**であるフィールドの変換には、時間がかかります（大規模なデータベースの場合数時間が必要です）。

この段階ではメッセージが表示されないため、変換が続行されているかどうかは明瞭ではありません。

変換状況を確認するには、変換コンピュータかデータベースサーバの活動状況を点検してください（CPUまたはI/O）。

- 11 変換ログファイル「sdu.log」を参照します。

警告:

AssetCenter 4.1.0またはそれ以降のバージョンを移行するユーザは、この注意事項を無視して結構です。

変換中にエラーが発生したら、

- 1 問題点のソースを訂正します。
- 2 「手順4 - 旧フォーマットの本番データベースのコピーを作成する」の段階から、変換を再開します。

変換に関する情報

警告:

AssetCenter 4.1.0またはそれ以降のバージョンを移行するユーザは、本項の内容を無視して結構です。

変換の際、以下に挙げるような規則が適用されます。

ヒント:

異なった動作を適用する場合は、対応する関連付けを「migration.xml」変換ファイル内で変更します。

旧フォーマットのシミュレーション用データベースの全バージョンで使用される規則

フロアプラン

[amFloorPlanPos] テーブルのレコードは変換時に削除されます。

データベースの構造パラメータ

変換プログラムは、選択された「カスタマイズ済4.2.0 gbbase.dbb」データベース記述ファイルで、定義されたテーブル、フィールド、リンクとインデックスのパラメータを適用します。

例えば、フィールドのデフォルト値の計算スクリプトの場合などです。

必須フィールド

ターゲットフィールドが以下の両方の条件を満たす場合、

- ターゲットフィールドが必須である。または、固有値を強制するインデックスの一部を成す。
- ターゲットフィールドが、明示的な関連付け（「migration.xml」変換ファイルで説明されている）や暗示的な関連付け（フィールドが同じSQL名である場合、自動的に推測される）の一部を成さない。

この場合、変換の第1段階で警告メッセージが発生します。

これは、データベース変換に先立つテスト段階です。

ユーザが変換を中断しない限り、変換は停止しません。

変換の中断は、データベースを変更する前に実行してください。データベースを変更した場合は、**旧フォーマットのシミュレーション用データベース**を復元してください。

旧フォーマットの本番データベースで、必須フィールドに必要な情報を入力しておくことを推奨します。

フィールドのデフォルト値

データベース構造で定義されたデフォルト値は適用されません。

デフォルト値が適用されるようにするには、値を変換ファイル内に定義する必要があります。

ヒント:

標準の変換ファイル「migration.xml」は、このタスクを実行する属性「value」を含んでいます。

固有値のインデックス

変換プログラムは、固有値インデックスの値が実際に固有であるかどうかを検証することはありません。

しかし、変換操作がインデックスの整合性に害を及ぼす場合、DBMSは変換を中断します。

value属性のSQLの有効性

変換プログラムは、「value」属性のSQLの有効性を検証しません。

しかし、SQLの観点から無効なvalue属性が存在する場合、DBMSは変換を中断します。

「グローバル」な方法での変換

変換作業は、ほぼすべてのデータに対して「グローバルな」方法で実行されます。レコード別に変換するのではなく、グローバルなSQLステートメントがテーブルのレコードすべてを変更します。

変更されたテーブル

テーブルを変更する場合（テーブル「A」）、変換ツールは以下の順番でタスクを実行します。

- 1 テーブル「A」の名前を変更します（例：テーブル「AOld」）。
- 2 新しいテーブルが作成されます（例：テーブル「A」）。
- 3 デフォルトでは、テーブル「AOld」からテーブル「A」にデータが転送されます。

Mapping要素は、別の動作を定義できます。

- 4 テーブル「AOld」が削除されます。

テーブルAは以下のように変換されます。

テーブルAは旧バージョン内に存在する	テーブルAは4.2.0バージョン内に存在する	旧バージョンと4.2.0バージョン間で、フィールド、リンクまたはインデックスが変更されている	変換プログラムの操作
はい	はい	いいえ	テーブルAで直接操作する
はい	はい	はい	中間テーブルAOldを作成する
いいえ	はい	無関係	新規テーブルAを作成する
はい	いいえ	無関係	テーブルAのデータを別のテーブルへ転送し、変換後にテーブルAを削除する

ヒント:

From属性はテーブルAOldを参照する必要はありません（テーブルAを参照するだけで十分です。変換プログラムは、いつAOldで情報を検索するべきか自動的に探知します）。

反対に、Mapping要素外で実行されるスクリプトでは、AとAOldの区別をする必要があります。

 **注意:**

変換されていないテーブルと、削除されたテーブルの名前は、データベース変換中に変更しません。

手動で変換するアプリケーションデータを格納するフィールド

手動で変換するアプリケーションデータを格納するフィールドは、変換ファイル内で定義された指令に従って空にされます。

デフォルトでインストールされる変換ファイル「migration.xml」は、空にされるフィールドが出力されるアプリケーションデータに対応するように書かれています。

4.0.0とそれ以降のバージョンの旧フォーマットシミュレーション用データベースに使用される規則

システムデータ

AssetCenterには、デモ用データベースまたは本番データベースにインポート可能なデータが付属しています。

- システムデータ：AssetCenterソフトウェアの基本機能に不可欠なデータ
システムデータは、バージョン4.0.0以降から、明確に識別されるようになりました。
ユーザはこれらのデータを変更できません。
- 専門分野データ：必要に応じて本番データベースへ挿入する基本データ
このデータは、機能ごとに分類されています。
- サンプルデータ：AssetCenterの使用に慣れるための便利なデータ

旧フォーマットの本番データベースのシステムデータは、変換時に4.2.0バージョンのシステムデータと自動的に一括して置換されます。

4.0.0より前のバージョンの旧フォーマットシミュレーション用データベースに使用される規則

属性

属性は資産のカテゴリから作成されます。

属性の名前に特に意味がないこともあります。

同じプロパティ（【属性】（SQL名：seNature）フィールドなど）のカテゴリが複数あると、対応するプロパティを含む属性1つのみが作成されます。

履歴

【amHistory】テーブルのレコードは変換されます。この履歴に含まれる情報は、レコードが旧フォーマットの本番データベースの一部であった時に変更された事項を、記述し続けます。

資産

以下のフィールドは、【amAsset】テーブルから【amComputer】テーブルへそのまま転送されます。

- ComputerDesc
- BIOSSource
- BIOSAssetTag
- dtBIOS
- ICPUNumber
- SoundCard
- VideoCard
- OSServiceLevel
- OSBuildNumber

旧フォーマットのシミュレーション用データベースがバージョン3.5.0またはそれ以前のバージョンであり、上記と同種の情報を含む任意管理項目が、転送される資産に関連付けられている場合、この任意管理項目が入力されていると、任意管理項目の値はフィールドの転送によって取得された値を上書きします。

任意管理項目のSQL名は以下の通りです。

- BiosMachine（ComputerDescフィールドと同等）
- BiosSource（BIOSSourceフィールドと同等）
- BiosAssetTagId（BIOSAssetTagフィールドと同等）
- BiosDate（dtBIOSフィールドと同等）
- ICPUCount（ICPUNumberフィールドと同等）
- SoundCardDescription（SoundCardフィールドと同等）
- GCard01Description（VideoCardフィールドと同等）
- OS01ServiceLevel（OSServiceLevelフィールドと同等）
- OS01BuildNumber（OSBuildNumberフィールドと同等）

💡 ヒント:

このタスクは、「migration.xml」ファイルの<PreActions>要素内で実行されます。
このタスクは、3.6.0またはそれ以降のバージョンの「migration.xml」では無効になっています。
この操作が必要な場合は、「migration.xml」ファイル内で対応する行をアクティブにします。

調整

変換時に、[amAdjustment] テーブルのレコードは [amPortfolio] テーブルへ転送されます。

4.2.0フォーマットのシミュレーション用データベースに負荷をかけないために、[amAdjustment] テーブルの以下のフィールドは削除されます。

- Name
- mTax*
- seAcquMethod
- IReqLinId
- IPOrdLinId
- IDelivLinId
- IInvLinId

また、ライセンス契約の調整は削除されています。

💡 ヒント:

これらの動作を変更する場合は、対応する関連付けを「migration.xml」変換ファイルに追加します。

消費

変換時に、[amConsUse] テーブルのレコードは [amPortfolio] テーブルへ転送されます。

この際、[amConsUse] テーブルの [mTax*] フィールドは削除されます。

💡 ヒント:

これらのフィールドに格納されている情報を保存する場合は、対応する関連付けを「migration.xml」変換ファイルに追加します。

製品明細

変換時に、【amProdCompo】テーブルのレコードは以下の方法で変換されます。

- 「標準構成」に対応するレコード（つまり、オプション【bSuppPackage】が「0」であるレコード）は、【amReqLine】テーブルへ転送されます。
- 「セット商品」に対応するレコード（つまり、オプション【bSuppPackage】が「1」であるレコード）は、【amCatProduct】テーブルへ転送されます。

【amProdCompo】テーブルに転送されたレコードの場合、【bInstantAssign】フィールドの値は自動的に「1」になります。

製品

全製品（【amProduct】テーブル）は【amModel】テーブルに転送されます。

以下の条件を1つでも満たす場合は、【amCatProduct】テーブルにも転送されません。

- 製品の【mPrice】フィールド値が「0」以外である。
- 製品は【amProdSupp】、【amPOrdLine】、【amDelivLine】または【amInvoiceLine】テーブルのレコードへリンクしている。

2製品P1とP2が【amCatProduct】テーブル内に作成される場合に、P2がP1のコンポーネントであり、またP1とP2の両方が【amPortfolio】テーブルへ転送されると、【amCatProduct】テーブル内に作成されるレコードの【bPreinstalled】フィールドの値は、「1」になります。

製品が【amProdSupp】、【amPOrdLine】、【amDelivLine】または【amInvoiceLine】テーブルのレコードへリンクしていると、製品は【mCatRef】テーブルへも転送されます。

インストール

テーブル【amProdSoftInfo】のレコードは、ライセンス製品（amProduct）とソフトウェア（amSoftware）間のリンクを確立します。

変換すると以下のテーブル内にレコードが作成されます。

- amCatProduct（セット商品に対応します）
- amReqLine（標準構成に対応します）

ライセンス契約

警告:

ライセンス契約の変換は、変換プロセスの中でも注意を要する部分です。
この変換プロセスは複雑です。
データベースでの最良のテスト方法は、標準の変換シミュレーションを実行した後で、結果の詳細を検証する方法です。

ライセンス契約は、[amContract] テーブルのレコードです。

- これらのレコードでは、[seType] フィールドの値は「5」です。
- レコードは、外部キー [ILicCntrlId] ([amAsset] テーブル) により、少なくとも1つの資産に関連付けられています。

この種の契約は以下の方法で変換されます (これは、簡略された説明です)。

- ライセンス契約はインストール済ソフトウェアに変換されます。この場合ライセンス契約は、[amPortfolio] テーブルへ転送され、[bSoftLicense] フィールドの値が「1」である属性に関連付けられているモデルへ、リンクされます。
- これらの契約にリンクしている [amWflInstance] テーブルのレコードは削除されます。
削除されたワークフローインスタンスへ関連付けられているレコードも、削除されます。
- 契約特有のフィールドやリンクの内、[amPortfolio] テーブルでは無意味なものは削除されます。
- これらの契約に関連付けられた資産の外部キー [ISoftLicUseRights] の値は、「0」になります。
- [seAcquMethod] フィールドの値は「0」になります。
- 契約と資産間のリンク ([AstCntrDescs] リンク経由で [amAstCntrDesc] テーブルへ格納されます) は、同じ資産上のインストール済ソフトウェア (amPortfolio) へ変換されます。
- 契約と従業員間のリンク ([Users] リンク経由で [amCntrEmpl] テーブルへ格納されます) は、ユーザアカウント (つまり、契約から [amPortfolio] テーブル内に作成されるライセンス下の従属ライセンス) に変換されます。
- 契約にリンクしていた [amAdjustment] テーブルのレコードは削除されます。
- 契約間の親子リンクは無くなります。

資産の任意管理項目

資産の任意管理項目は、変換時に資産に関連しているポートフォリオ品目に関連付けられます。例外は、任意管理項目が4.2.0データベースの1フィールドに

(特に【コンピュータ】(SQL名: amComputer) テーブルのフィールド) 転送される場合です。

あるフィールドに転送された任意管理項目は、資産から切り離されるため、関連するポートフォリオ品目には帰属しません。

ヒント:

変換ファイルにはコメントのMapping要素があり、これを有効にすると、上記の動作を変更できます。

見積

見積を基に作成された発注が全部または一部受領されている場合、変換時にこの見積は消失します。

他の見積は発注に転換されます。

競合の可能性

変換時に、新規の識別子(主キー)が作成されます(テーブル内に作成されるレコードごとに作成されます)。

しかしどのDBMSを使っている場合でも、データベースでの識別子数は 2^{31} に制限されています。

この数を超過すると最終データベースは破損します。

変換中に、エラーメッセージは警告を発生しません。

このため変換前に、この数を超過していないかどうかを確認する必要があります。

変換中に作成される識別子の最大数は、旧フォーマットのシミュレーション用データベースのバージョンにより異なります。

この数を超過していないことを確認するには、

- 1 旧フォーマットのシミュレーション用データベース内にある最大識別子(MaxIdと呼びます)の、おおよその値を決めます。

このためには、あるテーブル(【amLocation】など)にレコードを1つ作成し、このレコードの主キーの値(【amLocation】では、lLocald)を書きとめておきます。

ヒント:

主キーの値を表示するには、このフィールドをリストに追加します(【リストの設定】ポップアップメニュー)。

- 2 MaxIdが $(2^{31})/8$ よりも小さいことを確認してください。

 **注意:**

4.0.0バージョンの旧フォーマットシミュレーション用データベースでは、制約点はありません。

発注明細のメーカー

製品に関連する（ [Product] リンク ）発注明細の、 [メーカー]（ SQL名： Brand ）フィールドの値は、変換時に消失します（製品自体がメーカーに関連しているためです）。

他の発注明細の [メーカー]（ SQL名： Brand ）フィールドの値は、 [詳細]（ SQL名： LineDesc ）フィールドに追加されます。

依頼明細のメーカー

製品に関連する（ [Product] リンク ）依頼明細の、 [メーカー]（ SQL名： Brand ）フィールドの値は、変換時に消失します（製品自体がメーカーに関連しているためです）。

他の依頼明細の [メーカー]（ SQL名： Brand ）フィールドの値は、 [詳細]（ SQL名： LineDesc ）フィールドに追加されます。

6 4.2.0フォーマットのシミュレーション用データベースの整合性を検証する

1  **重要項目:**

旧フォーマットの本番データベースのバックアップコピーを作成します。

- 2 AssetCenter Database Administrator 4.2.0を起動します。
- 3 4.2.0フォーマットのシミュレーション用データベースに接続します（ [ファイル / 開く / 既存のデータベースを開く] メニュー）。
- 4 データベースの診断画面を表示します（ [アクション / データベースの診断 / 修復] メニュー）。
- 5 テーブルのリストで [(すべてのテーブル)] を選択します。
- 6 ログファイルの名前とパスを指定します。
- 7 [レコードの整合性のチェック] オプションを除くすべての検証オプションを選択します。
- 8 [解析のみ] オプションを選択します。
- 9 [実行] をクリックします。

- 10 実行画面のメッセージを確認します。
- 11 必要に応じて、ログファイルの内容を確認します。

 **ヒント:**

ログビューアを使用すると、ログファイルを閲覧できます。

プログラムが問題を表示する場合、以下の操作の内の1つを実行します。

- 1 変換ファイル「migration.xml」を変更します。
- 2 「**手順5 - 旧フォーマットのシミュレーション用データベースを変換する**」の段階から、操作を再開します。

 **警告:**

AssetCenter 4.1.0またはそれ以降のバージョンを移行するユーザは、上記の2手順を実行する必要がありません。

または、

- 1 旧フォーマットの本番データベースのデータを変更します。
- 2 「**手順4 - 旧フォーマットの本番データベースのコピーを作成する**」の段階から、操作を再開します。

分析 / 修復プログラムの詳細については、マニュアル『**管理**』の「**データベースの診断と修復**」の章を参照してください。

4.2.0フォーマットのシミュレーション用データベースの有効性を確定する

4.2.0フォーマットのシミュレーション用データベース内を検索して、変換が正常に行われたかどうかを確認します。

以下の操作などを実行することも可能です。

- 4.2.0フォーマットのシミュレーション用データベースと、旧フォーマットのシミュレーション用データベース間で、主要テーブル（資産、製品、従業員、発注など）のレコードの数を比較します。

レコード数が大幅に異なる場合は、正常に変換されているかどうか確認します。

レコード数が大幅に異なっても正常な場合：ライセンス契約は、変換時に契約のテーブルから削除されます。このため、契約のテーブルのレコード数が変換時に減少するのは普通です。

- 主要テーブルのレコードの詳細を確認し、情報に一貫性があるかどうかを確かめます。

契約の場合、契約のタイプごとに（リース、メンテナンスなど）レコードを点検します。

資産の【モデル】（SQL名：Model）リンクなどの微妙なリンクには、特に注意する必要があります。

- 任意管理項目とその値が変換されており、予測どおりに任意管理項目がフィールドに変換されているかどうか確認します（任意管理項目ごとに1テスト）。

問題点がある場合は、以下の操作の内の1つを実行します。

- 1 変換ファイル「migration.xml」を変更します。
- 2 「手順5-旧フォーマットのシミュレーション用データベースを変換する」の段階から、操作を再開します。
- または、
 - 1 旧フォーマットの本番データベースのデータを変更します。
 - 2 「手順4-旧フォーマットの本番データベースのコピーを作成する」の段階から、操作を再開します。

🔑 手順8 - 旧フォーマットの本番データベースでの権限の一部を制限する

旧フォーマットの本番データベースのユーザ権限を変更し、手動変換が必要なアプリケーションデータを含むテーブルを、ユーザが変更できないようにします。

- 1 本マニュアルの「参考情報」の章、「手動で変換するアプリケーションデータ」の節を参照して、どのアプリケーションデータを手動で変換すべきか判断してください。
- 2 ユーザ権限のリストを表示します（【管理/ユーザ権限】メニュー）。
- 3 各権限を1つずつ選択します。各権限ごとに以下の操作を行います。
 - 1 ユーザ権限に関連する全オブジェクトを選択します。
 - 2 【作成】、【削除】、【作成中に入力】権限をオフにします。
 - 3 【変更】をクリックします。

手動変換されるアプリケーションデータは、旧フォーマットの本番データベースから抽出されるため、この操作が必要になります。本番データベースをコピーした後に変更された事項は、変換時に取り込まれません。

🔑 手順9 - 手動で変換するアプリケーションデータを出力する

💡 注意

アプリケーションデータの詳細については、本マニュアルの「手動で変換するアプリケーションデータ」の章を参照してください。

🔧 手動で変換するアプリケーションデータを出力する

- 1 AssetCenter Database Administrator 4.2.0を起動します。
- 2 「Admin」ログインを使って、旧フォーマットの本番データベースに接続します（[ファイル/開く/既存のデータベースを開く]メニュー）。
- 3 [マイグレーション/アプリケーションデータの出力]メニューを選択します。
- 4 [ターゲットデータベースの記述ファイル]フィールドに、「カスタマイズ済4.2.0.gbbase.dbb」ファイルの完全パスを入力します（このファイルは、[マイグレーション/カスタマイズした構造の伝達]メニューの[生成フォルダ]フィールドで選択したフォルダ内にあります）。
- 5 [マイグレーションフォルダ]フィールドに値を入力します。これは、変換に必要な参照ファイルを含むフォルダです。
変換されるデータベースバージョンごとにフォルダがあります（「C:\Program Files\Peregrine\AssetCenter\migration\fromxxx」では、「xxx」が旧バージョンの番号に当たります）。
- 6 [作業フォルダ]フィールドで、手動変換し出力するアプリケーションデータの、保存用フォルダを指定します。
- 7 画面に表示されるメッセージを読みます。
- 8 ログファイル「sduxprt.log」を参照します（ファイルは、[作業フォルダ]フィールドで指定されるフォルダにあります）。

💡 ヒント:

ログビューアを使用すると、ログファイルを閲覧できます。

- 9 作成される「.xml」ファイルのツリー構造のバックアップコピーを作成します。
このコピーは、元の「.xml」ファイルを使用する場合、または「.xml」ファイルで変更された事項を表示する場合に便利です。

!! 出力中の規則

出力ツールは以下の操作を行います。

- 手動変換するアプリケーションデータのコピーを、手動変換が可能なフォーマットで出力します。
- アプリケーションデータだけでなく、アプリケーションデータのコンテキストに関する情報も出力します。これにより、AssetCenter Script Analyzerで簡単にアプリケーションデータを更新できるようになります。
- アプリケーションデータのタイプ別に構成された「.xml」ファイルのツリー構造を作成します。

各「.xml」ファイルは、確認される1つまたは複数のアプリケーションデータを含む1レコードに対応します。

- 旧フォーマットの本番データベースにユーザが追加したすべてのアプリケーションデータを包括します。
- システムデータを除外します。

システムデータは特別な処理を必要とします。本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」の章、「[手順5-旧フォーマットのシミュレーション用データベースを変換する](#)」の節の「[変換に関する情報](#)」を参照してください。

- このツールは、アプリケーションデータのテーブル、リンクおよびフィールドが、データベース4.2.0の構造と一致しているかどうかを確認しません。

💡 ヒント:

これは、AssetCenter Script Analyzerプログラムが実行します。

- **専門分野データとサンプルデータ**を含めます。

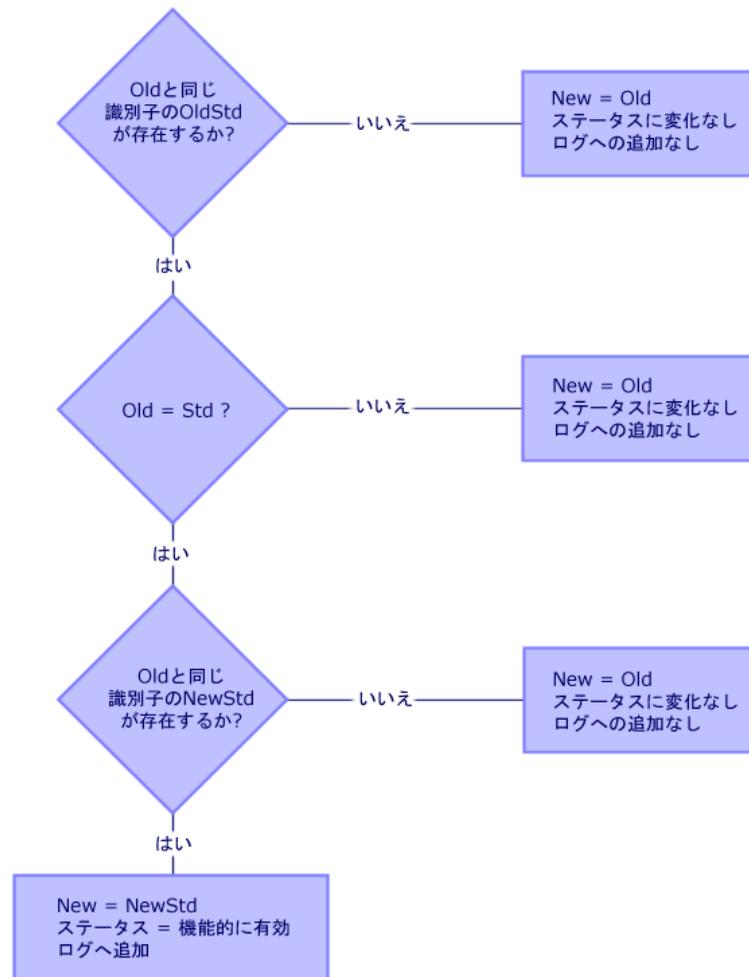
サンプルデータは特別な処理を必要とします。

この処理は、旧フォーマットの本番データベースにそのまま残しておいたアプリケーションサンプルデータを、更新するために実行されます。

この処理を実行するために、ツールは出力したアプリケーションデータを1つずつ検証します。

プロセスは以下の通りです。

図 5.2. サンプルデータの処理 - プロセス



説明

- Old : 出力されたアプリケーションデータ (つまり旧フォーマットの本番データベースから来るデータ)
- OldStd : Oldに一致する標準の旧フォーマットのサンプルデータ (サンプルデータが存在する場合)

OldとOldStdの比較は、データ型に基づく識別子を使って実行されます。例 : アクションでは【SQL名】 (SQL名 : SQLName) フィールド

ツールに参照されるアプリケーションのサンプルデータは、AssetCenter 4.2.0のインストール先フォルダの「/migration/fromXxx/reference」サブフォルダ内に格納されます。「Xxx」は、AssetCenterの旧バージョンの番号に当たります。

- NewStd : Oldに一致する標準の4.2.0フォーマットの**サンプルデータ** (サンプルデータが存在する場合)

OldとNewStdの比較は、データ型に基づく識別子を使って実行されます。

ツールに参照されるアプリケーションのサンプルデータは、AssetCenter 4.2.0のインストール先フォルダの「/migration/fromXxx/referencenew」サブフォルダ内に格納されます。「Xxx」は、AssetCenterの旧バージョンの番号に当たります。

- New : ツールに処理された後 (変更されているか、またはそのまま変更なし) のOld

手動で変換するアプリケーションデータを処理する

手動で変換されるアプリケーションデータは、数段階の手順を踏んで処理されます。

- 1 手順10 - アプリケーションデータを検証し、修正する
- 2 手順11 - 修正されたアプリケーションデータを復元する
- 3 手順12 - 4.2.0フォーマットのシミュレーション用データベースの整合性を検証する
- 4 手順13 - 復元されたアプリケーションデータを検証する

これらの手順は本節で説明されています。

注意:

「フィールド」の検証や変換が本節で説明されている場合、この「フィールド」は、AssetCenterデータベース構造のフィールドとリンク両方を指します。

ヒント:

アプリケーションデータの処理を、複数の従業員に分担作業させることも可能ですが、この場合は分担作業をプロジェクトとしての確に管理する必要があります。

10 手順10 - アプリケーションデータを検証し、修正する

このタスクはAssetCenter Script Analyzerを使って実行します。

アプリケーションデータの検証と修正

以下の手順に従います。各手順の詳細については、後述のAssetCenter Script Analyzerの情報を参照してください。

- 1 AssetCenter Script Analyzerを起動します。
- 2 **【作業フォルダ】**フィールドに値を入力します。
下記の🔗を参照してください。
- 3 手順「[手順3 - 旧フォーマットの本番データベースの構造の変更事項を伝達する](#)」の段階で、伝達されないスクリプトを含む「.xml」ファイルのツリー構造を作成した場合は、以下の手順に従います。
 - 1 「[手順3 - 旧フォーマットの本番データベースの構造の変更事項を伝達する](#)」の段階で作成した「<生成フォルダ>dbbscripts」フォルダと、「<生成フォルダ>builddbbscripts」フォルダをコピーします（フォルダが存在する場合）。
 - 2 構造の変更事項の伝達については、本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）](#)」の章、「[手順3 - 旧フォーマットの本番データベースの構造の変更事項を伝達する](#)」の節を参照してください。
 - 3 このフォルダを、**【生成フォルダ】**フィールドで指定したフォルダ内に貼り付けます。
- 4 検証するアプリケーションデータのリストを表示します（**【アクション/すべてのファイルのリスト】**または**【アクション/未処理のファイルのリスト】**メニュー）。
【メッセージ】ウィンドウに、検証する「.xml」ファイルのリストが情報と共に表示されます。

下記の🔗を参照してください。

アプリケーションデータを出力する際に、AssetCenter Database AdministratorはSQL名を「.xml」ファイルへ自動的に割り当てています。デフォルトでこの名前は、接頭辞と自動的に増分される数字から構成されています。

場合によっては、明確な名前も使用されています。

テーブルのSQL名	「.xml」ファイルに名前を付けるために使用されるフィールドのSQL名
amAction	SQLName

テーブルのSQL名	「.xml」ファイルに名前を付けるために使用されるフィールドのSQL名
amQuery	SQLName
amCalcField	SQLName
ItemNo	amFieldAdjustTempl
ItemNo	amFieldAdjust
OptSection	amOption

- 5 「手順3-旧フォーマットの本番データベースの構造の変更事項を伝達する」の段階で自動的に伝達されなかったスクリプトを、現段階で処理する場合は、これらのスクリプトに対応する「.xml」ファイルから処理します。
 - 1 「<生成フォルダ>/dbbscripts」フォルダと「<生成フォルダ>/builddb/dbbscripts」フォルダの、最初の「.xml」を選択します。
 - 2 ファイルを詳しく分析します（[アクション/スクリプトの問題を表示]メニュー）。
 - 3 [メッセージ]ウィンドウの内容を確認します。
下記の¹³と¹⁴を参照してください。
 - 4 「手順3-旧フォーマットの本番データベースの構造の変更事項を伝達する」の段階で取得した「カスタマイズ済4.2.0.gbbase.dbb」ファイル内で、対応するスクリプトを変更するために、AssetCenter Script Analyzerが提示する変更方法を使用します。
このためには、AssetCenter Database Administratorを起動し、「カスタマイズ済4.2.0.gbbase.dbb」ファイルを開いてスクリプトを手動で変更します。
 - 5 「.xml」ファイルの処理が終了したら、[機能的に有効]オプションを選択します。
 - 6 検証するアプリケーションデータのリストを新規に表示します（[アクション/処理するファイルを表示]メニュー）。
[メッセージ]ウィンドウに、検証する「.xml」ファイルのリストが表示されます。
 - 7 検証する次の「.xml」ファイルを選択し、このファイルを詳しく解析します。
- 6 要約のリストから、検証する別の「.xml」ファイルを選択します。
選択した各「.xml」ファイルごとに、
 - 1 ファイルを詳しく分析します（[アクション/スクリプトの問題を表示]メニュー）。

 ヒント:

1つの「.xml」ファイル内に、手動で変換すべきアプリケーションデータが複数存在する場合があります。

- 2 **[メッセージ]** ウィンドウの内容を確認します。
下記の  と  を参照してください。
- 3 「.xml」ファイルを、編集用ウィンドウの **[コンテキスト]** フィールドとタブで直接変更します。
変更された「.xml」ファイルは、変換プロセスで後ほどインポートされます。
下記の  を参照してください。
- 4 スクリプトをコンテキスト内でテストします（ **[アクション/スクリプトをコンテキスト内で確定]** メニュー）。
この操作は、スクリプトが、4.2.0データベース構造に対して有効であるかどうかを確認することを目的としています。

 重要項目:

無効なアクションスクリプトとSQLクエリをAssetCenterのグラフィカルインタフェースで開くことはできないため、この操作は不可欠です。また、「.xml」ファイルの復元後にアクションスクリプトを修正することは非常に困難であり、時には不可能になります。

この操作は、角括弧内のフィールドとリンクが、アクションのコンテキストに対して有効であるかどうかを検証します。

 注意:

いずれにせよ、 **[復元可能]** オプションを現在のファイル用に選択する際に、スクリプトはスクリプトのコンテキスト内で自動的にテストされます。

 警告:

スクリプトをコンテキスト内でテストしても、さらに **[アクション/スクリプトの問題を表示]** メニューを使ってテストする必要があります。このツールはスクリプトの他の側面をテストするためです。

- 5 「.xml」ファイル全体を検証し修正した後、 **[復元可能]** オプションを選択します。
これは、手動変換するアプリケーションデータをテストするために、「.xml」ファイルを、4.2.0 **フォーマットのシミュレーション用データベース** へ復元できることを意味します。
[復元可能] オプションを選択すると、スクリプトは自分のコンテキスト内で自動的にテストされます。
下記の  を参照してください。

- 6 検証するアプリケーションデータのリストを新規に表示します（[アクション/処理するファイルを表示]メニュー）。
[メッセージ]ウィンドウに、検証する「.xml」ファイルのリストが表示されます。
AssetCenter Script Analyzerは、「復元可能」となっている「.xml」ファイルを解析しません。括弧内の数は「0」です。青のアイコン  は、復元可能であることを示します。
- 7 検証する次の「.xml」ファイルを選択し、このファイルを詳しく解析します。

 **注意:**

一部のアプリケーションデータのステータスは既に [機能的に有効] になっています。

これは、「段階を追ってマイグレーションを実行する - シミュレーション (シミュレーション用データベース)」の章、「手順9-手動で変換するアプリケーションデータを出力する」/「出力中の規則」の節で説明されている段階で、サンプルデータから検出されたアプリケーションデータに当たります。

アプリケーションデータの修正速度を速める

 **警告:**

ここで説明されている方法を実行するには、細心の注意が必要です。
この種の作業に慣れていない場合や、自分で責任を持って作業を遂行できない場合には、実行しないでください。

一部の修正は繰り返し実行されます。

このため、全「.xml」ファイルに、[検索/置換]メニューを使用することも可能です。

以下の点には特に注意してください。

- [検索/置換]メニューを使用する前に、各段階ごとに「.xml」ファイル全体のバックアップコピーを作成します。
- 区切り文字も検索する文字列へ追加します。
- [完全に一致する単語]オプションを選択します。
- 置換の合計数を表示し、妥当であるかどうか確認します。
- 変換されたファイルの、置換前と置換後の相違点を分析します。

AssetCenter Script Analyzerのメニュー

表 5.3. AssetCenter Script Analyzer - メニュー

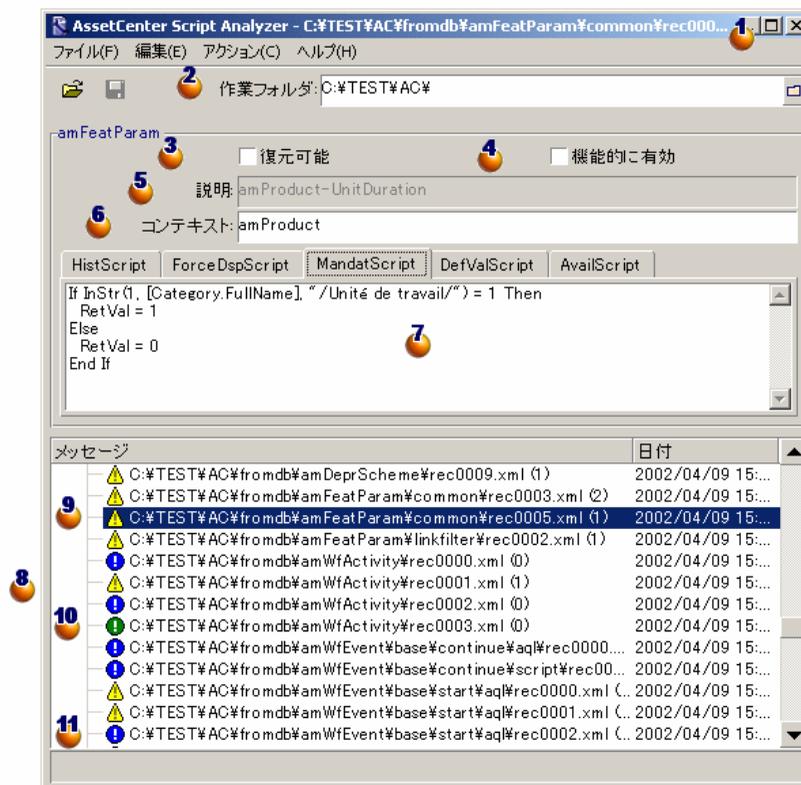
メニュー	機能
ファイルメニュー	
新規作成	特に用途はありません。
開く	【作業フォルダ】 フィールドで定義したルートを含むツリー構造の「.xml」ファイルを開きます。
保存	ファイルの変更事項を保存します（ 【復元可能】 または 【機能的に有効】 の属性、コンテキスト、スクリプト）。
名前を付けて保存	特に用途はありません。
終了	AssetCenter Script Analyzerを終了します。
編集メニュー	
他の編集メニューと同じ機能が含まれています。	
アクションメニュー	
次のファイルを開く	メッセージ ウィンドウに表示されたリスト内で、次の「.xml」ファイルを開きます。
前のファイルを開く	メッセージ ウィンドウに表示されたリスト内で、前の「.xml」ファイルを開きます。
スクリプトの問題を表示	選択された「.xml」ファイルの潜在的な問題を分析し、 メッセージ ウィンドウ内に結果を表示します。
スクリプトをコンテキスト内で確定	【コンテキスト】 フィールドに値が入力されている場合は、指定されたテーブルで現在のスクリプトの有効性をテストします。コンテキストが指定されていない場合は、コンテキスト外でスクリプトの有効性をテストします。

メニュー	機能
ファイルの復元可能な属性を強制	<p>【アクション/スクリプトをコンテキスト内で確定】メニューで、スクリプトがコンテキスト内で有効でないと判断されても、【復元可能】オプションが選択されます。</p> <p>警告:</p> <p>以下の場合にのみこのメニューを使用します。</p> <ul style="list-style-type: none"> 【アクション/スクリプトをコンテキスト内で確定】メニューが無効なエラーを返す場合。 しかし、スクリプトの有効性が明白である場合。
	<p>使用例</p> <p>【契約賃貸料】（SQL名：amCntrRent）テーブルには、【比例配分方法】（SQL名：ProrateField）フィールドがあります。このフィールドは、フィールドのシステム名を格納します。しかし、AssetCenter Script AnalyzerはSQL名の矛盾のみを検出します。このためAssetCenter Script Analyzerは、多くの場合根拠のないエラーを発生することになります。</p> <p>このフィールド用には以下の操作を実行します。</p> <ol style="list-style-type: none"> 【アクション/ファイルの復元可能な属性を強制】メニューを選択します。 ファイルを復元します。 ファイルをAssetCenter 4.2.0でテストします。 【比例配分方法】フィールドの値をAssetCenter Script Analyzer内で修正します。 【機能的に有効】オプションを選択します。
未処理のファイルのリスト	<p>「.xml」ファイルのリストを表示します。</p> <ul style="list-style-type: none"> 【作業フォルダ】フィールドで定義したルートを含むツリー構造の「.xml」ファイルが表示されます。 オプション【機能的に有効】が選択されていない「.xml」ファイルが表示されます。
すべてのファイルのリスト	<p>【作業フォルダ】フィールドで定義したルートを含むツリー構造の、全「.xml」ファイルのリストを表示します。</p>
アプリケーションデータの復元	<p>AssetCenterデータベースへの接続が選択可能になり、【復元可能】オプションがオンになっている「.xml」ファイルをインポートできるようになります。このメニューの機能は、AssetCenter Database Administratorの【マイグレーション/アプリケーションデータの復元】メニューの機能と同じです。</p>

AssetCenter Script Analyzerが表示する「.xml」ファイルのリスト

【アクション/すべてのファイルのリスト】または【アクション/未処理のファイルのリスト】メニューを使用すると、AssetCenter Script Analyzerで表示されるウィンドウは以下のようになります。

図 5.3. AssetCenter Script Analyzer - 「*.xml」ファイルの分析用ウィンドウ



1 「.xml」ファイルの完全パス

2 AssetCenter Database Administratorを使って出力されるアプリケーションデータを含むフォルダ（手動で変換するアプリケーションデータを含む「.xml」ファイルのツリー構造）。

これは、AssetCenter Database Administratorの **[マイグレーション/アプリケーションデータの出力]** メニューの **[作業フォルダ]** フィールドで指定したフォルダに当たります。

このフォルダのルートには、「modifications.xml」ファイルがあります。

このファイルは「migration.xml」変換ファイルから生成されています。

「modifications.xml」ファイルは、ソースデータベースの各フィールドへ適用され得る変換事項のリストです（フィールドを1つずつ列挙しています）。

「modifications.xml」ファイルは、フィールド名の問題点を診断するためにAssetCenter Script Analyzerのみに使用されます。

3 現在の「.xml」ファイルを分析、修正した後に、**[復元可能]** オプションを選択します。

4 AssetCenterデータベースに復元された「.xml」ファイルのアプリケーションデータの機能をテストした後、**[機能的に有効]** オプションを選択します。

5 この情報は、検証するべきアプリケーションデータを識別するために使用されます。この情報（例えば、アプリケーションデータを格納するレコードのSQL名など）は、AssetCenter Database Administratorでアプリケーションデータを出力する際に抽出されます。

6 アプリケーションデータのコンテキストのテーブル（コンテキストが存在する場合）。

! **警告:**

[アクション/スクリプトの問題を表示] メニューはこの情報をテストしません。コンテキストが有効であるかどうか（例えば、4.2.0バージョンで削除されたテーブルなど）は、ユーザが自分で調べる必要があります。

7 スクリプトに複数のスクリプトが含まれている場合、各スクリプトは別々のタブに表示されます。選択されているタブのスクリプトに問題がある場合（「modifications.xml」ファイル内のフィールド）、**[アクション/スクリプトの問題を表示]** メニューがメッセージを表示します。必要に応じてスクリプトを変更します。

8 **[作業フォルダ]** フィールドで定義されているルートを含むツリー構造の「.xml」ファイルが、リストとして表示されます。**[アクション/すべてのファイルのリスト]** メニューでは全ファイルが列挙され、**[アクション/未処理のファイルのリスト]** メニューでは、**[機能的に有効]** オプションがオフのファイルのみが表示されます。

9 このリストの各行は、1つの「.xml」ファイルに相当します。

括弧内の数は、検証するフィールドを含む「.xml」ファイルの行数に当たります。

数が「0」で、行がアイコン  で始まる場合、フィールドのSQL名は正常ですが、ファイルに間違ったアプリケーションデータが含まれています。アプリケーションデータのコンテキストを定義するテーブルを考慮に入れた上で、間違ったデータであると判断されるため、この場合リンクが不正である可能性が多大です。

数が「0」で、行が青いアイコン  で始まる場合、フィールドのSQL名は正常です。また、アプリケーションデータのコンテキストを定義するテーブルを考慮に入れた上で、ファイルには間違ったデータが含まれていないと判断されます。ファイルをAssetCenterデータベースへ復元し、テストすることができます。

 **注意:**

クリックするとファイルが開きます。

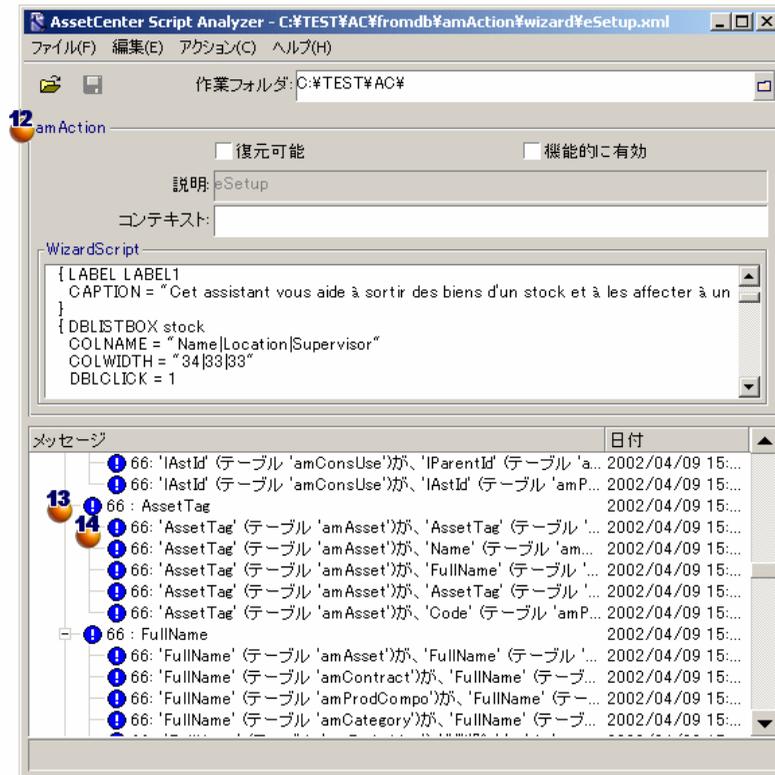
 緑色のアイコン  は、「.xml」ファイルのステータスが **[機能的に有効]** であることを意味します。

 青色のアイコン  は、「.xml」ファイルのステータスが **[復元可能]** であることを意味します。このステータスは、ユーザにより手動で選択される場合と、**[アクション / すべてのファイルのリスト]** と **[アクション / 未処理のファイルのリスト]** メニューにより自動的に選択される場合があります（「.xml」ファイルのフィールドが、「modifications.xml」内に存在しない場合に、コンテキスト内でスクリプトが確定されると、自動的に選択されます）。

スクリプトに関する問題点のリスト

[アクション / スクリプトの問題を表示] メニューを使用すると、AssetCenter Script Analyzerで表示されるウィンドウは以下ようになります。

図 5.4. AssetCenter Script Analyzer - スクリプトの分析用ウィンドウ



注意:

【メッセージ】枠で分析されるのは、現在のスクリプトのみです（複数のスクリプトが存在する場合は、現在のタブ）。

 「.xml」ファイルのアプリケーションデータを含むテーブルのSQL名

 問題のあるスクリプトの行番号と、「modifications.xml」ファイル内で検索されたフィールドのSQL名。

フィールドのSQL名が「modifications.xml」変換ファイル内にある場合は、これらのフィールド等を検証する必要があります。

解析プログラムは、フィールドやリンクの所属先テーブルを考慮に入れません。「modifications.xml」ファイル内にフィールドのSQL名が表記されるだけで、解析が必要であるフィールドであると見なされます。

SQL名は、テーブルA内で変更されていないフィールドのSQL名であると同時に、別のテーブルB内で変更されたフィールドのSQL名でもあり得ます。プログラムはこのような問題を検証し、必要があれば修正するようにユーザを手助けします。

「modifications.xml」ファイルのフィールド、リンクのSQL名を検索する際、英数字以外の全ての文字（例外：_）は、区切り文字と見なされます。

 **注意:**

ダブルクリックすると、カーソルは問題のある行の上に移動します。

14 下の各行は変更の提案事項に当たります。

このウィンドウには、検証されるフィールドのSQL名の修正方法ごとに、1行が表示されます。

行の始めにある数字は、「.xml」ファイル内で検証するべき行の数に当たります。

提示される修正方法は、「modifications.xml」変換ファイルで説明されている関連付けに基づきます。

提案事項は、「modifications.xml」ファイルで検索された関連付けに基づきます。

メッセージには複数のタイプがあります。

- 「'A' (テーブル'B') が、'C' (テーブル'D') に変更されました。」：スクリプトのフィールドAは、ソースデータベースのテーブルBの一部を成しています。フィールドAは、「modifications.xml」ファイル内でフィールドCへ関連付けられており、フィールドCは、「カスタマイズ済4.2.0 gbbase.dbb」ターゲット構造内でテーブルDの一部を成しています。

例：「'script' (テーブル'amAction') が、'memScript' (テーブル'amAction') に変更されました。」

- 「'A' (テーブル'B') はありません。」：スクリプトのフィールドAは、ソースデータベースのテーブルBの一部を成しています。フィールドAまたはテーブルBは、「カスタマイズ済4.2.0 gbbase.dbb」ターゲット構造の一部を成していません。
- 「'A' (テーブル'B') が、'C' (テーブル'D') に (式'E') 変更されました。」：スクリプトのフィールドAは、ソースデータベースのテーブルBの一部を成しています。フィールドAは、「modifications.xml」ファイル内でフィールドCへ関連付けられており、フィールドCは「カスタマイズ済4.2.0 gbbase.dbb」ターゲット構造内でテーブルDの一部を成しています。フィールドCは、式Eから入力されます。式Eは「modifications.xml」ファイル内にあります。Value属性がフィールドのSQL名と異なる場合に、式がメッセージに表示されません。

例（架空）：「'dtEnd'（テーブル'amTicket'）が、'duration'（テーブル'amTicket'）に（式'dtEnd - dtStart'）変更されました。」

 **注意:**

ダブルクリックすると、カーソルは問題のある行の上に移動します。

 **警告:**

問題のあるテーブル名の変更方法は提案されません。

手順11 - 修正されたアプリケーションデータを復元する

 **注意:**

変更されたアプリケーションデータの復元は、Connect-Itが実行します。Connect-Itがインストールされていると、この作業はバックグラウンドで実行されます。

アプリケーションデータの復元は、AssetCenter Database AdministratorまたはAssetCenter Script Analyzerで実行されます。

修正されたアプリケーションデータをAssetCenter Database Administratorで復元する

- 1 AssetCenter Database Administrator 4.2.0を起動します。
- 2 「Admin」ログインを使って、4.2.0フォーマットのシミュレーション用データベースに接続します（[ファイル/開く/既存のデータベースを開く]メニュー）。
- 3 [マイグレーション/アプリケーションデータの復元]メニューを選択します。
- 4 [マイグレーションフォルダ]フィールドに値を入力します。これは、変換に必要な参照ファイルを含むフォルダです。
変換されるデータベースバージョンごとにフォルダがあります（「C:\Program Files\Peregrine\AssetCenter\migration\fromxxx」では、「xxx」が旧バージョンの番号に当たります）。
- 5 [作業フォルダ]フィールドに値を入力します。これは、AssetCenter Database Administratorで出力されるアプリケーションデータを含むフォルダです（手動変換されるアプリケーションデータを含む「.xml」ファイルのツリー構造です）。

これは、AssetCenter Database Administratorの [マイグレーション/アプリケーションデータの出力] メニューの [作業フォルダ] フィールドで指定したフォルダに当たります。

- 6 [OK] をクリックします。
- 7 画面に表示されるメッセージを読みます。
- 8 ログファイル「sdurest.log」を参照します（ファイルは、[作業フォルダ] フィールドで指定されるフォルダにあります）。

ヒント:

ログビューアを使用すると、ログファイルを閲覧できます。

修正されたアプリケーションデータをAssetCenter Script Analyzerで復元する

- 1 AssetCenter Script Analyzerを起動します。
- 2 [作業フォルダ] フィールドに値を入力します。これは、AssetCenter Script Analyzerで修正されるアプリケーションデータを含むフォルダです（修正されるアプリケーションデータを含むXMLファイルのツリー構造です）。
- 3 [アクション/アプリケーションデータの復元] メニューを選択します。
- 4 4.2.0フォーマットのシミュレーション用データベースに、「Admin」ログインでアクセスします。
- 5 [マイグレーションフォルダ] フィールドに値を入力します。これは、変換に必要な参照ファイルを含むフォルダです。
変換されるデータベースバージョンごとにフォルダがあります（「C:\Program Files\Peregrine\AssetCenter\migration\fromxxx」では、「xxx」が旧バージョンの番号に当たります）。
- 6 [作業フォルダ] フィールドに値を入力します。これは、AssetCenter Database Administratorで出力されるアプリケーションデータを含むフォルダです（手動変換されるアプリケーションデータを含む「.xml」ファイルのツリー構造です）。
これは、AssetCenter Database Administratorの [マイグレーション/アプリケーションデータの出力] メニューの [作業フォルダ] フィールドで指定したフォルダに当たります。
- 7 [OK] をクリックします。
- 8 画面に表示されるメッセージを読みます。
- 9 ログファイル「sdurest.log」を参照します（ファイルは、[作業フォルダ] フィールドで指定されるフォルダにあります）。

 **ヒント:**

ログビューアを使用すると、ログファイルを閲覧できます。

 **拒否の原因**

- 「.xml」フォルダ内にあり、復元不可能と宣言されたアプリケーションデータは、拒否されます。
 - 4.2.0バージョンの全必須フィールドは、次のうち1つの条件を満たす必要があります。1、必須フィールドは、「modifications.xml」ファイル内のMapping要素である。2、必須フィールドは、旧バージョンから変更されていないテーブルの一部を成す。3、「modifications.xml」ファイルのMapping要素に関連付けられる2テーブル間で、必須フィールドのSQL名の1つが変更しない。
-

 **ヒント:**

フィールドが必須であるかどうかは、AssetCenter Database Administratorの[必須]パラメータで指定できます([はい]または[スクリプト]を選択します)。

手順12-4.2.0フォーマットのシミュレーション用データベースの整合性を検証する

本マニュアルの「段階を追ってマイグレーションを実行する - シミュレーション(シミュレーション用データベース)」章の「4.2.0フォーマットのシミュレーション用データベースの整合性を検証する」の節の指示に従って、4.2.0フォーマットのシミュレーション用データベースの整合性を検証します。

旧フォーマットの本番データベースへ接続するかわりに、4.2.0フォーマットのシミュレーション用データベースに接続します。

[修復]オプションのかわりに[解析のみ]を選択します。

プログラムが問題点を表示する場合、変換が適切に実行されなかった可能性があります。

問題点が表示されたら、特に「migration.xml」変換ファイル内の変換パラメータを確認します。

👉 手順13 - 復元されたアプリケーションデータを検証する

復元可能なアプリケーションデータとは、ユーザがAssetCenter Script Analyzerで確認し、必要に応じて変更したデータのことを指します。

復元可能でも、これらのアプリケーションデータがAssetCenterで正常に機能するとは限りません。

アプリケーションデータが正常に機能するかを確認するには、各データを手動でテストする必要があります。

- 1 復元された「.xml」ファイルを1つずつ開きます。
- 2 復元されたアプリケーションデータを含むレコードを検索します。
- 3 4.2.0フォーマットのシミュレーション用データベース内で検索されたアプリケーションデータをテストします。

💡 ヒント:

データベース構造の再編成が、手動変換されるアプリケーションデータを含むレコードの一般的な機能に、影響を与えないようにする必要があります（要素を修正するだけでは不十分です。例えば、現バージョンでは[ポートフォリオ品目]のテーブルが追加されているため、[資産]のテーブルを使用していたワークフローチャートなどを再検討する必要があります）。

🔪 注意:

スクリプトの параグラフの最後を示す記号は、文字「|」に置換されます。これによって、スクリプトの実行時に問題が起こることはありません。

- 4 復元されたアプリケーションデータ全体をテストした後、AssetCenter Script Analyzerの[機能的に有効]オプションを選択します。
これは、「.xml」ファイルを4.2.0フォーマットの移行データベースへ復元できることを意味します。

⚠️ クエリ - 注意事項

クエリが主キーの値に関連するあるレコードを識別していた場合に、このテーブルのレコードが変換時に再インデックス化されて新規テーブルへ移行されると、クエリは正しいリンクを選択できなくなります。

以下のいずれかの方法で修正します。

- クエリの主識別子を変更します。
- データベースの次回の移行時に同じ問題が生じる可能性があるため、クエリがより安定したフィールド値を使用するように変更します。

🔑 手順14 - 外部アプリケーションとの統合を調整する

旧フォーマットの本番データベースを外部アプリケーションと統合している場合は、これらのアプリケーションの統合モードを調整しなければならない場合もあります。

関連する可能性のあるアプリケーションについては、本マニュアルの「[段階を追ってマイグレーションを実行する - 最終変換（移行データベース）](#)」の章、「[手順20 - 4.2.0フォーマットの移行データベースを完成させる](#)」の節の、以下の部分を参照してください。

- [AssetCenter Web](#)
- [Get-It](#)
- [Get-Resources](#)
- [Connect-Itのシナリオ](#)
- [インポートスクリプト](#)
- [出力スクリプト](#)

「[手順20 - 4.2.0フォーマットの移行データベースを完成させる](#)」の後に、新規統合モードをアプリケーションに適用します。

しかし、マイグレーションの現段階でこの操作を準備しておく必要があります。これにより、この操作にかかる時間を節約することができます。

6 | 段階を追ってマイグレーション を実行する - 最終変換（移行 データベース）

この段階では、以下のファイル等が作成されています。

- 「カスタマイズ済4.2.0 gbbase.dbb」ファイル
詳しくは、本マニュアルの「段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）」の章、「手順3 - 旧フォーマットの本番データベースの構造の変更事項を伝達する」の節を参照してください。
- シミュレーション用データベースでテストされた「migration.xml」変換ファイル
- 手動変換され、4.2.0フォーマットのシミュレーション用データベースでテストされたアプリケーションデータ

本章では、4.2.0フォーマットの本番データベースを取得するための操作について、段階を追って説明します。

15 手順15 - 旧フォーマットの本番データベースの整合性を検証する

本マニュアルの「段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）」章の「手順1 - 旧フォーマットの本番

データベースの整合性を検証する」の節の指示に従って、旧フォーマットの本番データベースの整合性を検証します。

16 手順16 - 旧フォーマットの本番データベースをブロックしコピーを作成する

旧フォーマットの本番データベースをブロックする場合、旧フォーマットの本番データベースの使用を禁止して、変換中にデータベースを変更できないようにします（変更事項が適用されない可能性があるためです）。

以下の操作を行います。

- 1 すべてのユーザの旧フォーマットの本番データベースへの接続を解除します。
- 2 以下のプログラムを終了します。
 - AssetCenter Server
 - AssetCenter API
 - 旧フォーマットの本番データベースにアクセスする外部データベース
- 3 旧フォーマットの本番データベースへのアクセスをブロックします。
- 4 「段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）」の章の「手順4 - 旧フォーマットの本番データベースのコピーを作成する」で説明されている手順に従って、旧フォーマットの本番データベースのコピーを作成します。

旧フォーマットの本番データベースのこのコピーは、「移行データベース」と呼ばれます。

旧フォーマットの本番データベースのブロック時間は、ユーザの業務の妨げにならないよう最小限に抑えるようにします。

このため、実際に変換を実行する前にシミュレーションを念入りに行う必要があります。

17 手順17 - 旧フォーマットの移行データベースを変換する

旧フォーマットの移行データベースを変換するには、「段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）」の章、「手順5 - 旧フォーマットのシミュレーション用データベースを変換する」の節の「旧フォーマットのシミュレーション用データベースを変換する」の手順に従ってください。

- 旧フォーマットのシミュレーション用データベースへ接続するかわりに、旧フォーマットの移行データベースに接続します。
- シミュレーション用データベースで仕上げた「migration.xml」変換ファイルを使用します。

変換中は旧フォーマットの本番データベースをブロックするため、旧フォーマットの移行データベースの変換時間はできる限り短くするようにします。

シミュレーションを実行したにもかかわらず、予期せぬ問題に直面した場合は、以下の操作を実行してください。

- 1 旧フォーマットの移行データベースの変換を中断する。
- 2 ブロックされた旧フォーマットの本番データベースの使用を再開する。
- 3 新規の旧フォーマットのシミュレーション用データベースでシミュレーションを再度実行する。
- 4 「手順16-旧フォーマットの本番データベースをブロックしコピーを作成する」からマイグレーションプロセスを再開する。

18 手順18 - 手動で変換されたアプリケーションデータを復元する

4.2.0フォーマットの移行データベースで手動変換されたアプリケーションデータを復元するには、「段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）」の章、「手動で変換するアプリケーションデータを処理する」の節、「手順11 - 修正されたアプリケーションデータを復元する」の項の手順に従ってください。

- 4.2.0フォーマットのシミュレーション用データベースへ接続するかわりに、4.2.0フォーマットの移行データベースに接続します。
- 4.2.0フォーマットのシミュレーション用データベースで修正した「.xml」ファイル（作業フォルダ内）を使用します。

19 手順19 - 4.2.0フォーマットの移行データベースで整合性を検証する

本マニュアルの「段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）」章の「4.2.0フォーマットのシミュレーション用データベースの整合性を検証する」の節の指示に従って、4.2.0フォーマットの移行データベースの整合性を検証します。

旧フォーマットの本番データベースへ接続するかわりに、4.2.0フォーマットの移行データベースに接続します。

👉 手順20-4.2.0フォーマットの移行データベースを完成させる

以下の理由から、4.2.0フォーマットの移行データベースを修正する必要があります。

- 変換プログラムにより変換されなかったデータがある。
4.2.0フォーマットの移行データベースの一部のデータをテストし、手動で修正する必要があります。
- 追加または改善された機能がある。
追加または改善された機能を最大限に利用するには、4.2.0フォーマットの移行データベースで準備操作を行う必要があります。
これは、AssetCenterの性能とサービスを改善する良い機会でもあります。

完成作業（旧フォーマットの本番データベースの全バージョン用）

変換の成功の確認

変換が適切に実行されたかどうかを確認する必要があります。

確認するには以下の方法があります。

- 4.2.0フォーマットの移行データベースに目を通し、異常がないか点検します。
- 一部のテーブルのレコード数を、変換前と変換後で比較します。
レコード数が違う場合、「migration.xml」変換ファイルで意図的に指定されている場合と、異常な変換が原因である場合があります。

ストアドプロシージャup_GetCounterValの変更

本節の内容は、旧フォーマットの本番データベースでup_GetCounterValストアドプロシージャを変更したユーザを対象としています。

旧フォーマットの本番データベースを変換する前に以下の操作を実行します。

- 1 【amCounter】テーブルから別のテーブルへ派生されたカウンタを手動で変更します。

- 2 up_GetCounterValストアドプロシージャを、初期状態に戻します。
以下のテクノートの指示に従って、up_GetCounterValストアドプロシージャを新規に調整します。
 - Microsoft SQL Server : TN317171736
 - Sybase Adaptive Server : TN941931
 - Oracle Workgroup Server : TN12516652
 - DB2 UDB : TN1029175140

トリガ、インデックス、ストアドプロシージャ、ビュー

これらの要素に適用された変更事項は、旧フォーマットの本番データベースで変換前に初期状態に戻されています。

現段階では、必要に応じて変更事項を手動で新規に適用することができます。

フィールドのヘルプ

フィールド（とリンク）のヘルプは [フィールドのヘルプ]（SQL名：amHelp）テーブルに格納されています。

旧フォーマットの本番データベースの変換時に、このテーブルの内容は変更されません。

旧バージョンのフィールドのヘルプのカスタマイズ情報を保存する

- 1 フィールドのヘルプをそのまま出力します。
 - 1 AssetCenter 4.2.0を起動します。
 - 2 変換された4.2.0フォーマットの移行データベースに接続します（ [ファイル/データベースに接続] メニュー）。
 - 3 [フィールドのヘルプ] テーブル（ [管理/画面一覧] メニュー）のレコードリストを表示します。
 - 4 以下のフィールドとリンクを表示できるようにリストを設定します。
 - テーブル（TableName）
 - フィールド（FieldName）
 - 説明
 - 例
 - 注
 - 5 リストの内容を出力します（ [リストの出力] ポップアップメニュー）。
- 2 旧バージョン用標準のフィールドのヘルプを出力します。
 - 1 空のSybase SQL Anywhereデータベースを作成します。

空のシェルを作成する方法については、マニュアル『**管理**』の「AssetCenterデータベースの作成」の章、「DBMSを使って空のAssetCenterデータベースを作成する」/「DBMSレベルでデータベースを作成する」/「Sybase SQL Anywhere Runtime」の節を参照してください。

- 2 旧バージョンのAssetCenterを起動します。
 - 3 空のデータベースに接続します（**[ファイル/データベースに接続]**メニュー）。
 - 4 **[フィールドのヘルプ]**テーブル（**[管理/画面一覧]**メニュー）のレコードリストを表示します。
 - 5 以下のフィールドとリンクを表示できるようにリストを設定します。
 - テーブル（TableName）
 - フィールド（FieldName）
 - 説明
 - 例
 - 注
 - 6 リストの内容を出力します（**[リストの出力]**ポップアップメニュー）。
- 3 出力された2ファイルを比較します。
2ファイルの相違点は、ユーザが実行した変更事項に相当します。
変更事項に関する情報を保存しておいてください。

4.2.0バージョンのフィールドのヘルプを更新する

- 1 AssetCenter Database Administratorを起動します。
- 2 **[ファイル/開く]**を選択します。
- 3 **[データベース記述ファイルを開く(新規データベースの作成)]**オプションを選択します。
- 4 AssetCenter 4.2.0のインストール先フォルダの「config」サブフォルダの「標準4.2.0 gbbase.dbb」ファイルを選択します。
- 5 **[アクション/データベースの作成]**を選択します。
- 6 4.2.0フォーマットの移行データベースを選択します（**[データベース]**フィールド）。
- 7 **[データベースの作成]**オプションをオフにします。
- 8 **[システムデータの作成]**オプションをオフにします。
- 9 **[AutoCAD統合の使用]**オプションをオフにします。
- 10 **[フィールドのヘルプの使用]**オプションを選択します。
- 11 **[インポートシナリオの使用]**オプションをオフにします。
- 12 **[作成]**をクリックします。

フィールドのヘルプのカスタマイズ情報を再適用する

4.2.0バージョンのフィールドのヘルプを更新すると、カスタマイズ情報は削除されます。

保存した旧バージョンフィールドのヘルプのカスタマイズ情報に基づいて、手動で再度カスタマイズすることができます。

例えば、**[テーブル]** (SQL名: TableName) と **[フィールド]** (SQL名: FieldName) を識別キーとして使用して、変更事項をインポートすることができます。

恣意的に入力されたフィールド

変換時に妥当な情報がない場合、恣意的に値が入力されるフィールドもあります。

これらのフィールドに値を入力する方法は、「migration.xml」変換ファイル内に定義されています。

変換後にこれらのフィールドを簡単に検索できるようにするために、フィールドに入力されるデータベースの値には、文字「^」が付きます。

関連する全レコードで、これらのフィールド値を確認し、必要に応じて変更します。

関連するレコードが多数ある場合は、変更すべきレコードを出力した後インポートする、という方法を取ることもできます。

問題になるのは主に以下のフィールドです。

- **コード** (Code)
- **バーコード** (BarCode)
- **SQL名** (SQLName)
- **完全名** (FullName)
- その他

確認すべきフィールドの完全なリストを作成するには、

1 変換用に使用された「migration.xml」ファイルを、テキストエディタで開きます。

2 文字「^」を検索します。

これで、確認する全フィールドを検索できます。

例

```
<Mapping to="amAssetRent" from="amAssetRent">
  <Field sqlname="Code" value="'^' || SDUSTR IAssetRentId"/>
</Mapping>
```

この例では、**[amAssetRent]** テーブルのレコードの **[コード]** フィールドの値が、文字「^」で始まる場合、フィールド値を検証しなければなりません。

AssetCenter Web

AssetCenter Webを 4.2.0バージョンに更新する必要があります。

AssetCenter Webの標準ページのみを使用していた場合は、この操作で十分です。これにより、AssetCenter Webの新規の標準ページを使用するようになります。

追加Webページを作成した場合、または標準Webページをカスタマイズした場合は、以下の手順に従います。

- 1 追加ページまたはカスタムページを保存します。
- 2 AssetCenter Webを4.2.0バージョンに更新します。
- 3 各Webページをテストし適応するように変換します。

出カイベント

変換時に [amOutputEvent] テーブルのレコードは変更されません。

レコードの値は、旧フォーマットの本番データベースの構造を反映している可能性があります。

この場合、手動で変換を完成させる必要があります。

注意:

変換時に [amInputEvent] テーブルのレコードは変更されません。

しかし [amOutputEvent] テーブルの場合と異なり、この場合問題はありませ

Get-It

Get-Itで開発されたWebアプリケーションがAssetCenter 4.2.0データベースで機能するようにするには、

- 1 AssetCenter 4.2.0の動作環境の表 (Peregrine Systemsのカスタマサポート用Web サイト参照) に、使用中のGet-Itバージョンがあるかどうかを確認します。
- 2 必要に応じてGet-Itを更新します。
- 3 各カスタムWebページをテストし、適応するように変換します。

Get-Resources

Get-ResourcesがAssetCenter 4.2.0データベースと機能するようにするには、

- 1 AssetCenter 4.2.0の動作環境の表 (Peregrine Systemsのカスタマサポート用Web サイト参照) に、使用中のGet-Resourcesバージョンがあるかどうかを確認します。
- 2 必要に応じてGet-Resourcesを更新します。

- Get-Resourcesの標準ページのみを使用していた場合は、この操作で十分です。これにより、Get-Resourcesの新規の標準ページを使用するようになります。追加Webページを作成した場合、または標準Webページをカスタマイズした場合は、以下の手順に従います。
- 1 追加ページまたはカスタムページを保存します。
 - 2 必要に応じてGet-Resourcesを更新します。
 - 3 各カスタムWebページをテストし、適応するように変換します。

Connect-Itのシナリオ

- Connect-Itを使って4.2.0フォーマットの移行データベースにアクセスするには、AssetCenter 4.2.0付属のConnect-Itバージョンを使用しなければなりません。Connect-Itの既製シナリオを使用していた場合、移行後は新規の既製シナリオを使用します。
- 独自のシナリオを作成した場合は、
- 1 既製シナリオ以外の旧シナリオを保存します。
 - 2 Connect-Itを更新します。
 - 3 Connect-Itシナリオを1つずつ開きます。
 - 4 各シナリオで、以下の操作を行います。
 - 1 Connect-Itシナリオを開く際に警告メッセージが表示される場合は、メッセージを確認します。
 - 2 警告メッセージに応じてシナリオを訂正します。
 - 3 テスト用データを使ってシナリオを実行します。
 - 4 テスト中に問題が発生する場合は、問題点を訂正します。

インポートスクリプト

- 旧バージョンで作成したスクリプトを維持する場合、以下のインポートスクリプトを1つ1つテストする必要があります。
- 1 AssetCenter 4.2.0を起動します。
 - 2 テスト用データベースに接続します（テスト用データベースは4.2.0フォーマットの移行データベースのコピーでも構いません）。
 - 3 インポートモジュールを起動します（**[ファイル/インポート]**）。
 - 4 **[データベースのインポート]** オプションを選択します。
 - 5 **[テキスト]** タブを選択し、**[開く]** をクリックします。
 - 6 スクリプトを開きます（**[ファイル/スクリプトを開く]** メニュー）。
 - 7 各関連を1つずつ確認します（右側のリストで「ソース/ターゲット」の組をダブルクリックします）。
 - 8 変更事項を保存します（**[ファイル/保存]** メニュー）。
 - 9 インポートをテストします（**[インポート]** ボタン）。

10 必要に応じてインポートスクリプトを訂正する。

出力スクリプト

作成した出力スクリプトを保存する場合は、各スクリプトをテストする必要があります。

- 1 AssetCenter Export 4.2.0を起動します。
- 2 4.2.0**フォーマットの移行データベース**に接続します（出力しても、接続先のデータベースを変更することはありません）。
- 3 出力スクリプトを開きます（**[ファイル/スクリプトを開く]**メニュー）。
- 4 各クエリを1つずつ確認します。
 - 1 上のリストからクエリを選択します。
 - 2 画面下の**拡大鏡**アイコンをクリックします。
 - 3 クエリが有効である場合は、警告メッセージは表示されません。
 - 4 クエリが無効である場合は、警告メッセージが表示されます。
 - 5 警告メッセージが表示される場合もされない場合も、クエリのパラメータが予想通りであるかどうかを確認する必要があります。この際データベース構造が変更したことを考慮に入れます（例えば、資産のテーブル内で検索した情報は、現在はポートフォリオ品目テーブルにある可能性があります）。
- 5 変更事項を保存します（**[ファイル/保存]**メニュー）。
- 6 出力をテストします（**[アクション/スクリプトの実行]**メニュー）。
- 7 必要に応じて出力スクリプトを訂正します。

4.2.0フォーマットの移行データベースと詳細画面をカスタマイズする

AssetCenter 4.2.0では、テーブル、フィールド、リンク、ボタンおよびタブを追加できるようになりました。

これらのアプリケーションデータを追加するために、マイグレーションを使用できます。

フィールドのカスタマイズの詳細については、マニュアル『**管理**』の「**データベースのカスタマイズ**」の章を参照してください。

書式

旧**フォーマットの移行データベース**の変換時に、書式は変更されません。

従って、データベース構造の変更は反映されません。

多数のテーブル、フィールド、リンクのSQL名が無効になる可能性があります。

各書式で、以下のテストを行います。

- 1 AssetCenter 4.2.0を起動します。
- 2 書式のリストを表示します（[ツール/レポート機能/書式]メニュー）。
- 3 各書式を1つずつ選択します。
 - 1 警告メッセージが表示される場合は、メッセージの内容に応じて書式を修正します。
 - 2 書式を印刷して確認します。
 - 3 必要に応じて、書式を変更します。

ビュー

ビューは、旧フォーマットの移行データベースの変換時には変更されません。従って、データベース構造の変更は反映されません。

ビューは、適用されたフィルタと表示する列を記憶するため、1つ1つのビューを表示して、適用されたフィルタと表示する列を確定していきます。

- 1 AssetCenterを起動します。
- 2 各書式を1つずつ選択します（[ツール/ビュー]メニュー）。
- 3 警告メッセージが表示される場合は、メッセージの内容に応じてビューを修正します。

ヒント:

必要に応じて新しいビューを作成します。

Crystal Reports レポート

レポートは、旧フォーマットの移行データベースの変換時には変更されません。従って、データベース構造の変更は反映されません。

多数のテーブル、フィールド、リンクのSQL名が無効になる可能性があります。

旧レポートを再利用する場合

- 1 AssetCenterを起動します。
- 2 レポートのリストを表示します（[ツール/レポート機能/レポート]メニュー）。
- 3 保存する必要がないレポートを削除します。
- 4 残りの各レポートをテストします。

各レポートごとに以下の操作を行います。

 - 1 レポートのコンテキストとなる画面を開きます（例：資産のリストまたは詳細画面）。
 - 2 レポートの印刷画面を表示します（[フィル/印刷]メニュー）。

- 3 **【タイプ】**フィールドで、テストするレポートのタイプを選択します。
 - 4 レポートを選択します。
 - 5 **【プレビュー】**をクリックします。
 - 6 警告メッセージが表示される場合は、メッセージの内容に応じてCrystal Reportsでレポートを修正します。
- 5 さらに、AssetCenter 4.2.0付属の新しい標準レポートをインポートする場合は、
新規レポートをインポートする前に、保存する古いレポートのSQL名を変更
します。

 **警告:**

名前を変更しないと、古いレポートは同SQL名の新規レポートに上書きされてしまいます。

旧レポートを再利用しない場合

- 1 AssetCenterを起動します。
- 2 レポートのリストを表示します（**【ツール/レポート機能/レポート】**メニュー）。
- 3 古いレポートをすべて削除します。

AssetCenter 4.2.0付属の標準レポートをインポートする

【サンプルデータ】のレポートを、4.2.0フォーマットの移行データベース内へインポートするには、

- 1 AssetCenter Database Administratorを起動します。
- 2 **【ファイル/開く】**を選択します。
- 3 **【データベース記述ファイルを開く(新規データベースの作成)】**オプションを選択します。
- 4 AssetCenter 4.2.0のインストール先フォルダの「config」サブフォルダの「標準4.2.0 gbbase.dbb」ファイルを選択します。
- 5 **【アクション/データベースの作成】**を選択します。
- 6 4.2.0フォーマットの移行データベースを選択します（**【データベース】**フィールド）。
- 7 **【データベースの作成】**オプションをオフにします。
- 8 **【システムデータの作成】**オプションをオフにします。
- 9 **【フィールドのヘルプの使用】**オプションをオフにします。
- 10 **【AutoCAD統合の使用】**オプションをオフにします。
- 11 **【インポートシナリオの使用】**オプションを選択します。

- 12 [インポートするデータリスト] から [Crystal Reportレポート] を選択します。
- 13 [作成] をクリックします。

ユーザ権限、アクセス制限、ユーザプロファイル

新しいテーブル、フィールド、およびリンクが新しいデータベース構造に追加されたため、ユーザ権限、アクセス制限およびユーザプロファイルを調整する必要があります。

各アクセス制限のクエリを確認しなければなりません。

新しいテーブルとフィールドを既存のユーザ権限とユーザプロファイルに追加し、必要に応じて新しい権限と制限を作成します。

フィールドに置換された任意管理項目

変換ツールは、任意管理項目の値をターゲットフィールドへコピーしました。しかし、変換ファイルのスクリプトを使って任意管理項目とその値を削除するように設定しなかった場合は、現時点で手動削除する必要があります。

リンク型の任意管理項目のかわりになるリンク

リンク型の任意管理項目の値を、4.2.0データベースのリンクへ転換する場合、リンクのターゲットテーブルが変換前と後で違うと、リンクにはデータが入力されません。

例：変換前に任意管理項目は資産のテーブルに関連付けられていましたが、任意管理項目を置換するリンクは、変換後にポートフォリオ品目のテーブルに関連付けられています。

この例では、資産の識別子は削除されており、新規のポートフォリオ品目に別の識別子が作成されています。

任意管理項目を置換するリンクが入力されていない場合、この種のリンクを含むレコードを検索するクエリを、変換後に実行する必要があります。

この種のリンクには、手動でデータを入力しなければなりません。

完成作業（旧フォーマットの本番データベースが4.0.0バージョンより前の場合）

単位

変換時に、【単位】（SQL名：amUnit）テーブルの【次元】（SQL名：Dimension）フィールドと【記号】（SQL名：Symbol）フィールドは、数種のソースから入力されています。

作成された値を確認し、必要に応じて調整します。

製品シリーズから作成されるモデル

変換時に、【amFamily】テーブルは【amBrand】テーブルへ転送されます。

【amBrand】テーブルの【Name】と【FullName】フィールドで、変換時に挿入されたレコードの値を確認する必要があります。

これらのレコードを識別するには、【Name】フィールドの値の中で、「^」を含むものを探します。

国

変換時に、【amBrand】テーブルの【FullName】と【Name】フィールドは数種のソースから入力されています。

作成された値を確認し、必要に応じて調整します。

メーカー、単位、国

バージョン4.0.0から、メーカー、単位と国は、【amBrand】、【amUnit】と【amCountry】テーブルへのリンクにより入力されます。リストデータにリンクしたフィールドが入力することはありません。

フィールドとリンクの変換時に、レコードは【amBrand】、【amUnit】と【amCountry】テーブル内に作成されます。

この方法で作成されたレコードの一部は、元のレコードとほぼ同一です。

リストデータで最近設定した規準に、値の一部が一致しない場合もあります。実際、リストデータの値を削除する場合に、データベースに既に格納されたレコードを削除された値と共に残しておくことも可能です。

例：H.P.とHewlett Packard

変換時に【Name】フィールドでこれらのテーブルのレコードを並べ替え、重複する値を削除することも可能です。

属性

[名前]と[コード]フィールド

変換時に、[属性] (SQL名: amNature) テーブルの [名前] (SQL名: Name) と [コード] (SQL名: Code) フィールドは数種のソースから入力されています。

作成された値を確認し、必要に応じて調整します。

インストール済ソフトウェアから作成される属性

属性Softwareのサブ属性は、使用方法に応じて自由に再編成できます。

注意:

属性Softwareは、モデル「インストール済ソフトウェア」を属性「Software」へ関連付けるために使用されます。

ソフトウェアから作成されるモデル

変換時に、[amSoftware] テーブルは [モデル] (SQL名: amModel) テーブルへ転送されます。

この方法で作成されたモデルは、[名前] (SQL名: Name) フィールドの値が「^amSoftware」であるルートモデルに従属します。

モデルが「^amSoftware」モデルに関連付けられているか確認します。

モデル「^amSoftware」の名前を変更できます。

ライセンス契約から作成される資産

本節の内容は、ライセンス契約を作成したユーザを対象としています。

変換時にライセンス契約は、「^amSoftLic」というモデルに関連付けられた資産に転換されています。

このモデルは、「^amSoftLic」という属性に関連付けられています。

この場合、以下の操作を行います。

- 1 「^amSoftLic」というモデルに関連付けられた全資産を検索します。
- 2 これらの資産が関連付けられるモデルとして、より妥当なものがあるかどうか調べます。
- 3 見つからない場合は、モデルと属性の名前を変更します。

場所

変換時に、**【サプライヤ】**（SQL名：amCompany）テーブルの住所は**【場所】**（SQL名：amLocation）テーブルへ移されています。

この方法で作成された場所は、**【名前】**（SQL名：Name）フィールドの値が「^amCompany」であるルート場所に従属します。

場所「^amCompany」にリンクされている場所を確認します。

必要に応じて^amCompany場所の名前を変更します。

予算

【amBudget】テーブルを【amBudgLine】テーブルへ関連付ける「migration.xml」ファイルのMapping要素をアクティブにした場合、以下のテーブルの一部のレコードは無秩序に作成されます。

- amBudget
- amPeriod
- amFYDivision
- amFinancialYear
- amBudgClass
- amBudgCenter
- amBudgLine
- amBudgetCategory

これらのテーブル内でレコードを整理します。

【名前】（SQL名：Name）フィールドが「^amBudget」である予算を検証します。

【名前】（SQL名：Name）フィールドが「^amBudgClass」である予算分類を検証します。

【名前】（SQL名：Name）フィールドが「^amBudgCenter」である予算センタを検証します。

【名前】（SQL名：Name）フィールドが「^amBudget」である予算を検証します。

作成された期間を、適切な期間区分で編成します。

注意:

変換時に、期間区分は作成されません。

期間区分でまとめられた期間全体が、会計年度に相当するかどうか確認してください。

予算から作成されるコストカテゴリ

変換時に、【amBudget】テーブルは【amCostCategory】テーブルへ転送されません。

同じ名前の予算は変換中に名前が変わります。これは、変換後に各コストカテゴリの名前が異なるようにするためです。

必要に応じて【Name】フィールドを変換します。

これらのレコードを検索するには、【Name】フィールドの値の中で、「^」を含むレコードを探します。

機能ドメイン

変換時に、【名前】（SQL名：Name）フィールドの値がコピーされ【SQL名】（SQL名：SQLName）フィールドに入力されます。

このように作成されたSQL名は、この種のフィールドの規格に合っていない可能性があります（英数字と文字「_」のみが許可されています）。

各SQL名を確認し、必要に応じて規格に合うように名前を変更します。

機能権限

変換時に、【amEmplDept】テーブルの以下のフィールドはデータベース構造から削除されています。

- bEstimRight
- bHDCloseTickRight
- bHdProceedRight
- bHdSaveCallRight
- bOrderRight

これらのフィールドの値は、4.2.0フォーマットの移行データベースのフィールドへは移行されていません。

この場合、以下の操作を行います。

- 1 旧フォーマットの本番データベースの従業員のうち、これらのフィールドに値が入力されている従業員を検索します。
- 2 削除されたフィールドと同じ機能を果たす機能権限を作成します。
- 3 上記で検索された従業員に、適切な機能権限を関連付けます。

カタログリファレンス

【カタログ】（SQL名：amCatalog）テーブルのレコードを確認します。

特に、【名前】（SQL名：Name）フィールドが「OffCatalog」になっている【カタログ】（SQL名：amCatalog）テーブルのレコードを検証します。

このレコードは、変換された【amPOrdLine】テーブルのレコードから作成されたリファレンス（【amCatRef】テーブル）を含みます。

ライセンス契約へ関連付けられていた任意管理項目

変換時に、ライセンス契約（amContract）の一部は、ポートフォリオ品目（amPortfolio）へ変換されています。

ライセンス契約変換のプロセスについては、「[段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）](#)」の章、「[手順5 - 旧フォーマットのシミュレーション用データベースを変換する](#)」の節の「[ライセンス契約](#)」を参照してください。

ライセンス契約を説明するために使用されていた任意管理項目が、【amContract】テーブルで使われなくなっている可能性もあります。

【amContract】テーブルを対象とする任意管理項目のパラメータ（amFeatParam）へリンクしている任意管理項目（amFeature）を検索します。

使用されなくなっている任意管理項目と、任意管理項目のパラメータを削除します。

発注

変換時にソースデータ内に正確な情報がないと、【amPOrd】テーブルのレコードの【seStatus】フィールドは、見積から発注が作成されている場合は「**見積済**」になり、その他の場合は「**発注済**」になります。

【amPOrd】テーブルで作成されているすべての発注のステータスを確認します。

リポジトリの再構成

AssetCenterのリポジトリを構造化するデータモデルは、根本的に改善されました。

データを好条件下で使用し、AssetCenterの新機能を活用するには、以下の点について確認する必要があります。

1 新しいデータモデルを理解する。

このためには、マニュアル『[ポートフォリオ](#)』の「[概要（ポートフォリオ）](#)」の章を参照してください。

2 以下のテーブルの内容を確認し、必要に応じて改善する。

- **【属性】**（SQL名：amNature）テーブル
- **【モデル】**（SQL名：amModel）テーブル
- **【メーカー】**（SQL名：amBrand）テーブル
- **【資産】**（SQL名：amAsset）テーブル
- **【ポートフォリオ】**（SQL名：amPortfolio）テーブル

- [製品] (SQL名: amCatProduct) テーブル
 - [カタログリファレンス] (SQL名: amCatRef) テーブル
 - [カタログ] (SQL名: amCatalog) テーブル
 - [依頼] (SQL名: amRequest) テーブル
 - [コンピュータ] (SQL名: amComputer) テーブル
 - [電話機] (SQL名: amPhone) テーブル
 - [インストール済ソフトウェア] (SQL名: amSoftInstall) テーブル
- 3 構造が変更したことにより、調達モジュールの使用方法にどのような影響が及ぶかを理解する。

 注意:

AssetCenter 4.2.0では、新たに「オーバーフローテーブル」という概念を取り入れて、特定のデータを周辺テーブルに記録します。例えば、スキャンツールを使って入力したポートフォリオ品目に関する情報は、オーバーフローテーブルに保存されます。オーバーフローテーブルの導入により、特定のフィールドが以下のオーバーフローテーブルに移動されます。

- [コンピュータ] (SQL名: amComputer) テーブル
- [電話機] (SQL名: amPhone) テーブル
- [インストール済ソフトウェア] (SQL名: amSoftInstall) テーブル

経費付替えと予算のトラッキング

バージョン4.0.0から、コスト会計のデータと予算トラッキングのデータの処理における可能性が広がりました。

データを好条件下で使用し、AssetCenterの新しい機能を利用するには、以下の操作を行います。

- 1 ファイナンスモジュールの新機能を理解する。
このためには、マニュアル『**ファイナンス**』の「**経費**」の章、「**経費の概要**」の節を参照してください。
- 2 ファイナンスモジュールに関連するテーブルの内容を確認し、細かく指定する。
ファイナンスモジュールに関連するテーブルの一覧については、マニュアル『**ファイナンス**』の「**参考情報**」の章、「**テーブル(ファイナンス)**」の節を参照してください。

7 | 段階を追ってマイグレーション を実行する - 最終段階

本章では、4.2.0フォーマットの移行データベースを再稼動するための操作について、段階を追って説明します。

AssetCenterプログラムを更新する

管理用コンピュータとクライアントコンピュータで、AssetCenterプログラムを更新する必要があります。

AssetCenterと共に使用するプログラムのバージョンが、AssetCenter 4.2.0と互換性があるかどうか確認します。必要に応じて、プログラムのアップグレードを実行します。

AssetCenterプログラムの一覧と、AssetCenterと共に使用するプログラムの一覧については、マニュアル『インストール』の「AssetCenterプログラムのリスト」の章を参照してください。

AssetCenter 4.2.0と互換性のあるプログラムのバージョンを確認するには、ペレグリンシステムズのカスタマサポートのサイトを参照してください。

 ヒント:

マニュアル『インストール』の「AssetCenterインストール後の操作」の章にも、互換プログラムのバージョンに関する一部の情報が記載されています。

AssetCenter Serverを管理用コンピュータにインストールする

AssetCenter Serverは、AssetCenterデータベースにおける全ての自動処理機能タスクを実行します。AssetCenter Serverが起動されていない場合、AssetCenterは正しく作動しません。

このため、以下の操作を行う必要があります。

- 1 AssetCenter Serverをクライアントコンピュータにインストールします。
- 2 AssetCenter Serverを設定します。
- 3 AssetCenter Serverを常時稼動にします。

AssetCenter Serverの使用については、マニュアル『管理』の「AssetCenter Server」の章を参照してください。

4.2.0フォーマットの移行データベースのAssetCenter キャッシュを削除する

4.2.0フォーマットの移行データベースへの接続と共にキャッシュを使用していた場合は、キャッシュを削除するようお勧めします。

キャッシュに関する詳細は、マニュアル『はじめに』の「参考情報」の章、「接続 / AssetCenterのパフォーマンス」の節を参照してください。

AssetCenterプログラムを更新する

プログラムを更新するには、AssetCenter 4.2.0をインストールします。

インストールプログラムは、AssetCenterの旧バージョンを検出します。旧バージョンが検出されると、旧バージョンを上書きするのか、または新しいフォルダにAssetCenterをインストールするのかを、確認するメッセージが表示されます。

注意:

上書きするとは、旧バージョンをアンインストールし、新バージョンをインストールすることを意味します。このオプションの唯一の制約点は、AssetCenter 4.2.0が旧バージョンのフォルダと同じフォルダにインストールされることです。

インストールプログラムがSQL Anywhereデータベース（拡張子が「.db」であるファイル）をインストールする際に、同じ名前のファイルがコンピュータに既存する場合は、このファイルを上書きするかどうかを選択します。

警告:

SQL Anywhereデータベースを保持する場合は、「上書き」を選択しないでください。

インストールの方法や注意事項については、マニュアル『インストール』を参照してください。

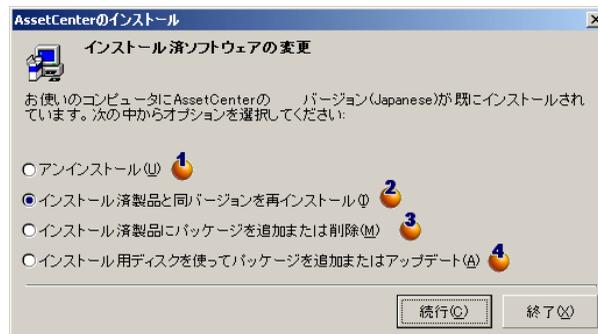
ヒント:

AssetCenter 4.2.0を変換用コンピュータにインストールする場合は、AssetCenterの旧バージョンをしばらく保存しておいてください。

インストールプログラムの画面

ここでは、インストールプログラムの主要画面を説明します。

AssetCenterの4.2.0バージョンが既にインストールされている場合は、以下の画面が表示されます。



1と**4**：これらのオプションは、更新時には使用しません。

2：前回のインストール内容に対して、コンポーネントの追加や削除を実行しない場合は、このオプションを選択します。

3：前回のインストール内容に対して、コンポーネントの追加や削除を実行する場合、また実行する可能性がある場合は、このオプションを選択します。このオプションは、クライアントコンピュータに特定のコンポーネント（AssetCenter Serverなど）をインストールしない場合に便利です。

上記の画面で [...パッケージを追加または削除] オプションを選択すると、以下の画面が表示されます。



1：既にインストール済みのコンポーネントが選択されています。追加するコンポーネントを選択します。再度インストールするコンポーネントは選択したままにし、アンインストールするコンポーネントのチェックボックスをオフにします。

2：このオプションを選択すると、既に選択され、インストールされているコンポーネントが再度インストールされます。このオプションをオフにすると、新しいコンポーネントのみがインストールされます。

**注意:**

上記の手順を自動化するには、コマンドラインからの自動インストールを使用します。詳しくは、マニュアル『インストール』の「AssetCenterプログラムのインストール」の章、「自動インストール(コマンドライン)」の節を参照してください。

AssetCenterが正常に起動することを確認する

AssetCenter 4.2.0を起動できない場合は、カスタマサポートまでお問い合わせください。

古い接続を削除して、新しい接続を作成する

この操作により、4.2.0フォーマットの移行データベースへ、ユーザが接続するようにします。

マニュアル『はじめに』の「参考情報」の章、「接続」の節を参照してください。

古い接続を変更することも可能です。

必要に応じて、接続用にAssetCenterキャッシュを作成します。

必要に応じてクライアントでAssetCenterのカスタマイズ設定を変更する

マニュアル『はじめに』の「ユーザのコンピュータでAssetCenterをカスタマイズする」の章を参照してください。

4.2.0フォーマットの移行データベースを実際に使用する

この段階は、マイグレーションプロセスの最終段階です。

この段階に入る前に、以下のことが既に実行されているはずです。

- 旧フォーマットの移行データベースが完全に変換され、4.2.0フォーマットの移行データベース修正されている。

- クライアントコンピュータと管理用コンピュータでAssetCenterプログラムが更新されている。

以下の操作を実行します。

- 1 AssetCenter Serverを、変換が完了した4.2.0フォーマットの移行データベース上で起動します。
- 2 外部プログラムを起動し、4.2.0フォーマットの移行データベースにアクセスします。
- 3 データベースが使用可能であることをユーザに通知します。

8 | 用語解説

マイグレーション

マイグレーションとは、AssetCenterの旧バージョンからバージョン4.2.0に移行するために実行する操作全体を指します。

マイグレーションでは、以下の操作を行います。

- 旧フォーマットの本番データベースを変換して（構造と内容）、AssetCenterのバージョン4.2.0と互換性があるようにする。
- 管理用コンピュータとクライアントコンピュータでAssetCenterプログラムをバージョン4.2.0にアップグレードする。

AssetCenterプログラムを更新する

AssetCenterのマイグレーションに必要な操作の1つです。

管理用コンピュータとユーザコンピュータでAssetCenterプログラムを再インストールし、プログラムのバージョン4.2.0を使用できるようにする。

次の用語と混同しないでください。

旧フォーマットの本番データベースを変換する

旧フォーマットの本番データベースを変換する

AssetCenterのマイグレーションに必要な操作の1つです。

旧フォーマットの本番データベースの変換とは、データベースの構造と内容を変更して、AssetCenterのバージョン4.2.0と共に使用できるようにすることを指します。

変換を行うには、複数の操作を実行する必要があります。手動で行う操作と、自動的に実行される操作があります。

次の用語と混同しないでください。

AssetCenterプログラムを更新する

変換ファイル

変換ファイルは、**旧フォーマットの本番データベース**の変換時にどのデータをどのように変換するのかを説明しています。

変換ファイルの名前は、「migration.xml」です。

変換ファイルは一般的に、「C:\Program Files\Peregrine\AssetCenter\migration\fromxxx」フォルダに格納されています。

AssetCenterは、デフォルトの変換ファイルと共にインストールされます（マイグレーションでサポートされる旧バージョンのAssetCenterにつき1ファイル）。

変換ファイルは、カスタマイズ可能なファイルです。

変換用コンピュータ

旧フォーマットの本番データベースを4.2.0形式に変換するために使用されるコンピュータ。

本マニュアルで指示されている通りに変換用コンピュータを設定する必要があります。

本番データベース

ポートフォリオを管理するために使用するAssetCenterデータベース。

次の用語と混同しないでください。

デモ用データベース

トリガ

データベースのフィールドまたはリンクが変更された場合に、AssetCenterが自動的に起動するアクション。

データ

フィールドを使ってデータベースに格納されるレコードの情報が、**データ**と呼ばれます。

アプリケーションデータ

「**アプリケーションデータ**」とは、本番データベースに（構造内にはなく）格納されているデータで、変換時に検証されなければならないデータを指します。

ヒント:

アプリケーションデータは、バージョン4.2.0で削除または変更された可能性のあるテーブル、フィールドまたはリンクを参照するため、確認する必要があります。

アプリケーションデータは以下のカテゴリに分類されます。

- Basicスクリプト
- AQLクエリ
- テーブル名を格納するフィールド
- フィールド名を格納するフィールド
- ウィザード
- 特殊文字列（コンテキスト内でのリンクとフィールドの連結）

データベースの構造

データベース構造は以下の集合を含みます。

- テーブル
- フィールド
- リンク
- インデックス

および上記要素のパラメータも含みます。以下のようなパラメータがあります。

- 記述文字列
- 有効性
- 妥当性
- 履歴保持
- 読取り専用
- 必須
- 無関連
- デフォルト値

これらのパラメータは、AssetCenter Database Administratorを使って固定値またはスクリプトで定義されます。

パラメータは、データベース記述ファイル「gbbase.dbb」内、またはデータベース内に保存されます。

9 | 参考情報

変換ファイル「migration.xml」の調整

警告

⚠ 警告:

変換ファイルを変更調整するには、高度な技術力と、AssetCenterの旧バージョンとバージョン4.2.0に関する深い知識が必要です。

変換ファイルの調整は、ペレグリンシステムズ公認のエンジニアが実行しなければなりません。

変換ファイルを非公認のユーザが変更する場合、変更者が責任を取ることになります。ペレグリンシステムズは一切責任を負いません。

💡 ヒント:

ペレグリンシステムズは、提携パートナー企業と共にマイグレーションをお手伝いします。経験豊富なコンサルタントが、変換ファイルの調整を担当します。

本節の情報は、公認エンジニアのみを対象としています。

 **重要項目:**

「migration.xml」変換ファイルをカスタマイズする前に、ファイルの名前や場所を変更してはなりません。これは、このファイルを使用するツールが、ファイルを標準フォルダ内で検索するためです。

このため、変換ファイルを変更する前に、バックアップコピーを作成しておくことをお勧めします。

参考情報

変換ファイルの定義に関しては、本マニュアルの「用語解説」の章、「変換ファイル」の節を参照してください。

変換ファイルを修正する時期については、本マニュアルの「段階を追ってマイグレーションを実行する - シミュレーション (シミュレーション用データベース)」の章、「変換ファイル「migration.xml」の調整」の節を参照してください。

変換ファイルの役割

変換ファイルは、以下の理由からフィールドの値を維持できない場合にフィールドの変換規則を定義するために使われます。

- フィールドの帰属先テーブルがなくなった、または帰属先テーブルのSQL名が変更した。
- フィールドが使用されなくなった、またはフィールドのSQL名が変更した。
- フィールドが、4.2.0フォーマットのデータベースのテーブルの直接フィールドへ転送される任意管理項目に属している。

 **注意:**

リンクは外部キー（フィールド）を使って処理されます。

変換ファイルは、旧フォーマットのデータベースの変換用SQLステートメントを生成するために使われます（DBMSに固有のSQL）。

変換規則

一部の变換規則は変換プログラムにより自動的に決定されます。

- AssetCenterの旧バージョンと4.2.0バージョン間でテーブルの構造が同じである場合（テーブルのSQL名、フィールドとインデックスが同じである場合）：

フィールドを変換ファイル内で宣言する必要はありません。フィールド値は変化しません。

ヒント:

必要であれば、構造が変化しないテーブルのフィールドとリンク用に変換を定義することもできます。

- フィールドのSQL名が、変換ファイルのMapping要素内で関連付けられるソーステーブルとターゲットテーブルで同一の場合：
これらのフィールドは自動的に関連付けられます。フィールド値を変えない場合は、これらのフィールドを変換ファイル内に記入する必要はありません。

変換ファイルのシンタックス

ファイルのグローバルシンタックス

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE MigrationFile SYSTEM "acmig.dtd">
<MigrationFile continueonerror=[AA]>
  <StartScript engine='G'>
    [A]
  </StartScript>
  <Translate table="[R]" into "[S]"/>
  <Mapping to="[C]" from="[B]" where="[K]" orderby="[O]" groupby="[P]" having="[Q]" autofill="[L]">
  <PreActions engine='T'>
    [U]
  </PreActions>
  <Field sqlname="[E]" value="[F]" translate="[X]" feature="[Y]" featuretable="[Z]">
    <Exception engine='M' value="[N]"/>
  </Field>
  <PostActions engine='V'>
    [W]
  </PostActions>
</Mapping>
<Script engine='O'>
  [I]
</Script>
<!--[J]-->
```

```
<!--[P]-->
</MigrationFile>
```

;?xml version="1.0" encoding="iso-8859-1"?>行

この行は必須です。

ファイルで使用されるXMLのバージョンと文字セットを指定します。

文字セットは変更可能ですが、「.xml」ファイルで使用される文字セットに対応しなければなりません。

<!DOCTYPE MigrationFile SYSTEM "acmig.dtd">行

この行は、どの「.dtd」ファイルをどの「.xml」ファイルへ関連付けるかを指定します。

AssetCenterは、「acmig.dtd」ファイルを「migration.xml」変換ファイルの隣にインストールします。

「acmig.dtd」は必須ではありませんが、「.xml」ファイルの構造を指定し、ファイルの読み取りを簡略化するためには便利です。

「acmig.dtd」を有効にするには、XMLエディタを使用する必要があります。

MigrationFile要素

この要素は、変換時に実行する操作を説明する3つの要素を含みます。

- StartScript
- Translate
- Mapping
- Script

continueonerror属性

この属性は任意です。

「AA」が「no」である場合、変換中に最初のエラーが発生した時点で変換は中断します。

「AA」が「yes」である場合、変換中にエラーが発生しても、可能な限り変換は続行します。

この属性のデフォルト値は「no」です。

engine属性

この属性は必須ではありませんが、要素の適用先DBMSを定義するために、複数の要素がこの属性を使用します。

以下の値を使用できます。

- Sybase
- MSSQL
- Oracle
- DB/2
- SQLAnywhere

大文字と小文字の区別をする必要があります。

SQLAnywhereでは完全版を使用します。ランタイムバージョンは、変換に必要なSQL命令をサポートしません。

StartScript要素

この要素は、**旧フォーマットのデータベースの変換前**（旧テーブルの名前を変更する前）に実行するSQLスクリプト [A] を含みます。

可能な限り、PreActions要素を使用することをお勧めします。これにより変換ファイルのメンテナンスが楽になります。

StartScript要素が有用な場合の例

- 複数のMapping要素が、同じPreActions要素を実行する必要がある場合
- **旧フォーマットのデータベースの構造のカスタマイズ情報を削除する場合**
- トリガを無効にする場合

スクリプトは、**旧フォーマットデータベースのDBMSのSQL言語**に対応したSQL言語で書かれなければなりません。

ヒント:

この制約点には例外が1つだけあります。文字列を結合するには、どのエンジンでも演算子「||」を使用しなければなりません（MSSQLとSybaseでは、これは「+」に変換されます）。

警告:

AssetCenterのAQL言語は認識されません。

SQL命令の各行は、GO行により実行されます。

例

```
UPDATE amPortfolio SET IParentId=0 WHERE IPortfolioItemId IN (SELECT p.IPortfolioItemId FROM amAssetOld a, amPortfolio p WHERE a.IParentId=0 AND p.IAstId=a.IAstId)
GO
DELETE FROM amItemListVal WHERE IItemId=(SELECT IItemId FROM a
```

```
mItemizedList WHERE Identifier='amBrand')
GO
```

engine属性

DBMSが[G]であるデータベースでStartScript要素が実行される際、engine属性付きのStartScript要素は、engine属性なしのStartScript要素のかわりになります。

Translate要素

この要素は、テーブル名（アクションのコンテキストなど）を格納するフィールドの変換時に使用されます。

ソーステーブル[R]が複数のターゲットテーブル[S]に、複数のMapping要素内で関連付けられる場合、Translate要素を定義する必要があります。

Translate要素では、テーブル名を格納するフィールドの自動変換時に、複数のテーブル[S]の内どれがターゲットテーブルになるのかを指定できます。

テーブル名を格納するフィールドの変換は、「migration.xml」変換ファイル内の情報を基に変換の実行時に自動作成される、マップテーブルを使用します。

マップテーブルは以下の要素を対応させます。

- Mapping要素内でto="[C]"属性とfrom="[B]"属性により関連付けられるテーブル（テーブル[C]と[B]が別である場合）
- Translate要素内でtable="[R]"属性とinto="[S]"属性により関連付けられるテーブル

Translate要素による関連付けは、Mapping要素による関連付けよりも優先されます。

UPDATEコマンドを使うと、変換ファイルのスキプトはマップテーブルを使用できるようになります。

これにより、テーブルの古い名前を新しい名前に変換できます。

例

```
UPDATE amDocument SET DocObjTable = ( SELECT newsqIname FROM sduTrans
WHERE oldsqlIname = amDocument.DocObjTable ) WHERE amDocument.DocObjTable
IN( SELECT oldsqlIname FROM sduTrans)
```

Mapping要素

この要素は、旧構造のテーブルのフィールドを、4.2.0バージョン構造のテーブルへ転送し変換します。

from属性

from属性は必須であり、旧構造のテーブル[B]を識別します。

結合の場合は、以下のシンタックスに従うと複数のテーブルを使用できます。

```
from="[テーブル1のSQL名] alias1, [テーブル2のSQL名] alias2, ..., [テーブルnのSQL名] aliasn"
```

to属性

to属性は必須であり、新構造のテーブル [C] を識別します。

where属性

where属性は任意の属性で、SQL条件 [K] を指定します。このSQL条件は、Mapping要素がテーブル [B] のどのレコードを処理するかを定義します。

where句はデフォルトで、ソーステーブル[B]のNULL主キーのレコードを除外します（内部結合 - where [主キーのSQL名] <> 0）。

where句は、[B]テーブルにリンクされているリモートテーブルのNULL主キーのレコードを、デフォルトで含みます（外部結合）。

例えば以下の関連付けでは、

```
<Mapping to="amCatProduct" from="amProdSoftInfo s, amSoftware soft" where="s.ISoftId = soft.ISoftId">
```

s.ISoftIdとsoft.ISoftIdが同じであるレコードは保持されます。

NULL主キーのレコードの役割については、マニュアル『AssetCenterの高度な使い方』の「AQLクエリ」の章、「AQLクエリの記述に関する推奨事項」の「主キーが0のレコードの理由と有用性」の節を参照してください。

orderby属性

orderby属性は任意の属性で、SQL並べ替え [O] を指定します。

groupby属性

groupby属性は任意の属性で、SQL部分集合（サブセット） [P] を指定します。

having属性

having属性は任意の属性で、SQL検索条件 [Q] を指定します。

autofill属性

autofill属性は任意の属性で、「yes」または「no」の値をとります。デフォルト値はyesです。

値がnoの場合、Field要素が処理する [C] テーブルのフィールドのみに値が入力されます。

変換プログラムが自動的に関連付けるフィールドには、値は入力されません（これらのフィールドのSQL名は、[B] テーブルと [C] テーブル内で同じです）。

PreActions要素

この要素は、Field要素を実行する前に、実行するSQLスクリプト [U] を含みません。

PreActions要素が有用な場合の例

- 変換するデータベースの内容から独立した属性を作成する場合
- 任意管理項目を作成する場合

この要素のシンタックスは、StartScript要素のシンタックスと同じです。

上級ユーザは、Mapping要素で実現できる操作を行うためにこの種のスクリプトを使用できます。

PreActions要素が実行される段階では、旧テーブルはまだ削除されていません。

このためこの段階ではまだ旧データを使用できます。

PreActions要素は特に、**旧フォーマットの本番データベース**の標準構造を変更したユーザを対象にしています。

Field要素

この要素は、SQL名 [E] の新フィールドへ、SQL式 [F] の計算値を入力します。

SQL式 [F] は、フィールドのSQL名で識別される [B] テーブルのフィールドを必要とします。

SQL式 [F] がDBMSで無効である場合、Field要素の行のすぐ後にException要素を入力する必要があります。

feature属性

ソース任意管理項目の値を、4.2.0フォーマットのデータベースの1フィールドに変換する際に、この属性を使用します。

この属性の値 [Y] は、変換する値を含む任意管理項目のSQL名に当たります。

featuretable属性

ソース任意管理項目の値を、4.2.0フォーマットのデータベースのフィールドに変換する際に、この属性を使用します。

この属性の値 [Z] は、変換する任意管理項目値を格納するテーブルのSQL名に当たります。

警告:

任意管理項目値を格納するテーブルと、任意管理項目の値は、[Z]テーブル内で関連付けられています。任意管理項目値を格納するテーブルは、Mapping要素のfrom属性で宣言されなければなりません。

例えば、[amFVAsset]テーブルは任意管理項目を格納します。この任意管理項目とレコードは、[amAsset]テーブル内で関連付けられます。あるフィールドの任意管理項目 [Y] の値を変換する場合、[amAsset]テーブルをfrom属性で宣言し、[amFVAsset]テーブルをfeaturetable属性で宣言する必要があります。

Exception要素

この要素は、DBMSのすぐ前のField要素用に、あるDBMSに専用の例外を作成します。

engine属性

engine属性は、例外が適用される [O] DBMSを定義します。
Exception要素は、[O] DBMSでField要素のかわりになります。

value属性

value属性は、[O] DBMS用に有効なSQL式を定義します。
結合の場合は、以下のシンタックスに従ってエイリアスを使用します。

```
value="[テーブルのエイリアス].[フィールドのSQL名]"
```

SDU_NEWID変数

この変数は、主キー用に新規の値を指定するvalue属性により、時折使用されません。

SDU_NEWIDは、旧フォーマットのデータベースの主キーの最大識別子の値に「1」を加えたものです。

SDU_NEWIDは変換プログラムにより自動的に計算されます。

PostActions要素

この要素は、Field要素を実行した後に、実行するSQLスクリプト [W] を含みません。

PostActions要素が有用な場合の例

- **【完全名】** (SQL名: FullName) フィールド値を計算する場合
- フィールドに転送された任意管理項目とその値を削除する場合

この要素のシンタックスは、StartScript要素のシンタックスと同じです。
上級ユーザは、Mapping要素で実現できる操作を行うためにこの種のスクリプトを使用できます。
PostActions要素が実行される段階では、旧テーブルはまだ削除されていません。
このためこの段階ではまだ旧データを使用できます。
PostActions要素は特に、旧フォーマットの本番データベースの標準構造を変更したユーザを対象にしています。

Script要素

この要素は、Mapping要素の実行後、および旧テーブルの削除前に実行するSQLスクリプト [I] を含みます。

可能な限り、PostActions要素を使用することをお勧めします。これにより変換ファイルのメンテナンスが楽になります。

Script要素が有用な場合の例

- 複数のMapping要素が、同じPostActions要素を実行する必要がある場合
- Mapping要素を使って整理作業を実行する場合
- 古いリストデータを削除する場合

この要素のシンタックスは、StartScript要素のシンタックスと同じです。

Script要素が実行される段階では、旧テーブルはまだ削除されていません。

このためこの段階ではまだ旧データを使用できます。

Script要素は特に、旧フォーマットの本番データベースの標準構造を変更したユーザを対象にしています。

!--要素

このタグを使うと、コード上にコメント [J] を挿入できます。コメントは変換プログラムに無視されます。

!----要素

このタグを使うと、変換ファイルのユーザ用にコメント [J] を挿入できます。コメントは変換プログラムに無視されます。

特殊文字を使用する

本節では、特殊な方法で解釈される一部の文字について説明します。

本節は全特殊文字を網羅していません。詳細については、SQLやXMLのマニュアルを参照してください。

一般的に、変換ファイルの一般構造はXMLの規則に従い、属性の値はSQLの規則に従います。

以下の文字は特殊な方法で解釈されます。

特殊文字	解釈	例	文字がテキストとして解釈される場合の対応文字	例
"	XML属性の値を区切る	value="!AssetRentId"	/	value="!/'"
'	属性の値の中で、SQLテキスト文字列を区切る	value="soft.Publisher+'/' +soft.Name"	"	value="''''"
<	XMLタグを開く	</Mapping>	<	value="<"
>	XMLタグを閉じる	</Mapping>	>	value=">"
&	エンティティの開始を示す	<	&	value="&"
;	エンティティの終了を示す	<	&を前につけない;	value=";"
/	SQLエスケープ文字		//	value="//"
	文字列連結のSQL演算子(全DBMSで有効)	value="'A' 'B'"	' '	value="'A' 'B'"

旧テーブルのフィールドを新規の複数テーブルへ配分する

これは例えば、旧資産テーブルのフィールドを、資産テーブルとポートフォリオ品目のテーブルへ配分する場合などです（旧バージョンの「資産」は2つの新規レコードを作成することになります）。

主識別子は、ポートフォリオ品目テーブル内に作成されなければなりません。これは、ポートフォリオ品目がAssetCenterデータベース内で（つまり1テーブル内だけでなく）固有でなければならないためです。

以下のタイプのMapping要素を作成する必要があります。

```
<Mapping to="amPortfolio" from="amAsset">
  <Field sqlname="lPortfolioItemId" value="SDU_NEWID+lAstId"/>
</Mapping>
```

任意管理項目をフィールドへ転送する

AssetCenter 4.2.0では、標準データベース構造の新規フィールドと、ユーザがカスタマイズした新規フィールドへアクセスできます。

旧フォーマットの本番データベース内で使用されている任意管理項目のかわりに、これらの新規フィールドを使用することも可能です。

この操作は、任意管理項目を広範囲にわたって使用している場合にのみ意義があります。

長所

- 詳細画面にある任意管理項目よりも自由にフィールドの位置を替えられます。
- アクセス制限は、任意管理項目よりもフィールド上で高性能を発揮します。

ヒント:

アクセス制限の性能は、リンクと任意管理項目上では同じです。

短所

- 任意管理項目のパラメータの [使用可能] (seAvailable) フィールドに相当するものは、フィールドにはありません。
- 任意管理項目とは違って、フィールドをクラスに関連付けることはできません。

シンタックス

```
<Mapping to="[ターゲットテーブルのSQL名]" from="[任意管理項目の値を格納するソーステーブルのSQL名]">
  <Field sqlname="[ターゲットフィールドのSQL名]" value="[任意管理項目の値を格納するフィールドのSQL名]" feature="[ソース任意管理項目のSQL名]" featuretable="[任意管理項目の値が格納されているテーブルのSQL名]" />
</Mapping>
```

全テーブルにエイリアスを使用します。任意管理項目の値を格納するフィールドを参照するvalue属性以外で、エイリアスを属性用に使用します。

Value属性は以下の値をとります。

- 任意管理項目がテキストを格納する場合は、ValString
- 任意管理項目が数値を格納する場合は、fVal
- 任意管理項目が日付を格納する場合は、dtVal

例

```
<Mapping to="amComputer A" from="amAsset">
  <Field sqlname="VideoCard" value="ValString" feature="Carte video" featuretable="amFVAsset" />
</Mapping>
```

制約点

任意管理項目をフィールドへ転送する方法には、以下の短所があります。

- 多数の結合を呼び出すことになる。
- 変換過程を遅らせる恐れがある。
- 任意管理項目の継承を管理できない。
- 転送された任意管理項目の値と、任意管理項目自体の値の削除を管理できない。

このタスクを実行するためには、Field要素の後に、PostActions要素を追加できます。

または、変換後に手動で削除します。

多数の任意管理項目を変換する場合は、以下の例のように<Script>要素を使用する方法が適しています。

```
UPDATE amComputer
SET ComputerDesc = (SELECT F.ValString
FROM amFVAsset F, amFeature V, amAsset A
WHERE IComputerId = SDU_NEWID * 2 + A.IAstId AND F.IFeatId = V.IFeatId AND V.SQLName='fv_BiosMachine')
```

```
GO
DELETE FROM amFVAsset WHERE IFeatValId IN ( SELECT IFeatValId FROM am
FVAsset F, amFeature V WHERE F.IFeatId = V.IFeatId AND V.SQLName='fv_BiosM
achine' )
GO
```

リンク型の任意管理項目で起こり得る問題点

リンク型の任意管理項目の値を、4.2.0データベースのリンクへ転換する場合、リンクのターゲットテーブルが変換前後で違うと、リンクにはデータが入力されません。

例：変換前に任意管理項目は資産のテーブルに関連付けられていましたが、任意管理項目を置換するリンクは、変換後にポートフォリオ品目のテーブルに関連付けられています。

この例では、資産の識別子は削除されており、新規のポートフォリオ品目に別の識別子が作成されています。

任意管理項目を置換するリンクが入力されていない場合、この種のリンクを含むレコードを検索するクエリを、変換後に実行する必要があります。

手動変換用アプリケーションデータを格納するフィールドを変換する

以下のようなMapping要素は、手動変換されるアプリケーションデータを格納するフィールドを、変換時に故意に空にします。

```
<Mapping to="amAccessRestr" from="amAccessRestr">
  <Field sqlname="ReadCond" value=""/>
</Mapping>
```

しかし、空にされたアプリケーションデータを含むレコードは、移行時に保存されます。

手動変換されるアプリケーションデータは消失しません。アプリケーションデータはAssetCenter Database Administratorを使って出力されており、変換プロセスの後半で復元されます。

手動変換すべき要素に対して、コンテキストの役目を果たさないテーブルの名前を格納するフィールドは、自動的に変換されます。

変換機能のパラメータ設定には、Translate要素を使用できます。

結合を使用する

結合は以下の規則に従わなければなりません。

- エイリアスは、結合されたテーブルそれぞれに定義されなければなりません。
- Field要素の属性where、orderby、groupby、havingとvalueの式は、エイリアスでテーブルを識別します。

警告:

結合が、**メモフィールド型**または**可変長バイナリフィールド型**のフィールドを対象とする場合、結合はサポートされません。

例

```
<Mapping from="amProdSoftInfo s, amSoftware soft" to="amCatProduct" where="s.ISoftId = soft.ISoftId">
  <Field sqlname="ICatProductId" value="s.IProdSoftId"/>
  <Field sqlname="InternalRef" value="soft.Publisher+'/'+soft.Name+'/'+soft.VersionLevel"/>
  <Field sqlname="FullName" value="+'/'+soft.Publisher+':'+soft.Name+':'+soft.VersionLevel+'"/>
  <Field sqlname="dtLastModif" value="s.dtLastModif"/>
</Mapping>
```

注意:

from 属性が指定する最初のテーブルには特別なステータスがあります。

このテーブルのフィールドが変換ファイルにない場合、このテーブルのフィールドは、ターゲットテーブルの同SQL名のフィールドと自動的に関連付けられます。

外部キーを使用する

外部キーは、複数のテーブルのレコード間のリンクを作成するために使用されます。

例

```
<Mapping from="amAsset" to="amPortfolio"
  <Field sqlname="IParentId" value="SDU_NEWID+IParentId"/>
</Mapping>
```

ソーステーブルを複数のターゲットテーブルに配分する

1ソーステーブルを2つ以上のターゲットテーブルに配分する場合、ターゲットテーブル内に作成される主識別子が、AssetCenterデータベース内で固有であるように保証する必要があります。

これには、以下のタイプのField要素を作成します。

```
<Mapping to="amPortfolio" from="amAsset">
  <Field sqlname="IPortfolioItemId" value="SDU_NEWID * 2 + IAsId"/>
</Mapping>
```

数値フィールドをテキスト文字列へ変換する

データの変換時には、数値文字列をテキスト文字列へ変換する必要が時折生じます。

例えば、数値タイプのフィールドを基準に、テキストタイプのフィールド値を計算する場合などです。

この変換をSQL言語で実行するのは複雑であり、また各エンジンにより変換方法は異なります。

提供されているマクロSDUSTRは、全タイプの数値フィールドの変換を全エンジン用に実行します。

例

```
<Mapping to="amPortfolio" from="amSoftInstall">
  <Field sqlname="Code" value="'^' || SDUSTR IInstId"/>
</Mapping>
```

この例では、

- [IInstId] フィールドは32ビット整数タイプです
- [Code] フィールドはテキストタイプです。
- SDUSTRマクロは、IInstIdをテキスト文字列に変換します。
- 変換された文字列には文字「^」が連結されます。
- 連結文字列は、[Code] フィールドへ挿入されます。

アプリケーションデータを手動で変換する

一部のMapping要素には、手動変換用アプリケーションデータを空にする役割があります。

例

```
<Mapping to="amAccessRestr" from="amAccessRestr">
  <Field sqlname="TableName" value=""/>
</Mapping>
```

空になったフィールドは、手動変換用アプリケーションデータの復元時に、新たに入力されます。

変換ファイルから生成されるSQL命令

変換ファイルは、旧フォーマットのデータベース（データと構造）の変換用にDBMSが使用するSQL命令を生成します。

例

以下のMapping要素

```
<Mapping from=[F] to=[T] where=[W]>
  <Field sqlname=[F1] value=[V1]/>
  <Field sqlname=[F2] value=[V2]/>
  ...
  <Field sqlname=[Fn] value=[Vn]/>
</Mapping>
```

は以下のSQLに相当します。

```
Insert Into T(F1; F2, ..., Fn)
Select V1 as F1, V2 as F2, ..., Vn as Fn
From A
Where W
```

変換ファイルを使用前に検証する

警告:

「acmig.dtd」ファイルに変換ファイルが適合するかどうかを、変換ファイルの使用前に確定する必要があります。

適合性を確認するには、Internet ExplorerまたはXMLエディタを使用します。

以下の内容のテストも推奨されています。

- 変換ファイルは、from、to、where、groupbyの組み合わせを1回のみ使用できる。
- Mapping要素は、ユーザがデータベースを使用する方法と一致している。

- 同じソース主キーから作成された複数の主キーは、それぞれ異なっている（SDU_NEWID変数の適切な使用）。
- 変換時に作成される主キーを格納する外部キーが、適切な主キーに対応している。
- 関連付けられていないソースフィールドとターゲットフィールド（変換ファイル内で手動で関連付けられておらず、また変換ツールにより自動的に関連付けられていない）が、意図的に無関連になっている。

これを検証するには、

- 1 変換のログフォルダにある「sdu.xml」ファイルを表示します。
 - 2 NotMappedSrcとNotMappedDstを検索します。
- where属性が定義するレコードの部分集合は、お互いに重複しない。部分集合の集まりがレコードの全体を占める。
 - テーブルが数回関連付けられても、複数のリンクの作成が不可能な場合（[IParentId]リンクや[ICommentId]リンク）、同じレコードに複数のリンクが作成されない。

特定のテーブルに関連していない資産の転送

場合によっては、ある一定のタイプの資産（コンピュータなど）を記述するための**オーバーフローテーブル**が存在します。

この場合、Mapping要素が変換ファイル内に用意されています。

特定の**オーバーフローテーブル**がない場合（車などの場合）、**旧フォーマットの本番データベース**内のデータの構造方法をそのまま保存しておくことをお勧めします。

標準データベースの構造と旧バージョンの構造の相違

AssetCenter 4.2.0は、以下のバージョン間における標準データベースの構造の相違点を記述するファイル（diff*.*）と共にインストールされます。

- バージョン4.2.0
- 旧バージョン

警告:

「diff*.＊」ファイルは、**旧フォーマットの本番データベース**で行われたカスタマイズ事項を考慮に入れません。

「diff*.＊」ファイルは、以下の形式で使用可能です。

- テキスト（「diff*.txt」ファイル）

- XML (「diff*.xml」ファイル)
- HTML (「diff*.htm」ファイル)

これらのファイルは一般的に、「C:\Program Files\Peregrine\AssetCenter\doc\Infos」フォルダに格納されています。

相違ファイルは、インストール時に「文書」コンポーネントを選択した場合にインストールされます。

相違ファイルの名前は、以下のように指定されています。

diff<AssetCenterの旧バージョン>.*

ヒント:

バージョン番号を確認するには、AssetCenterソフトウェアの旧バージョンを起動して、[ヘルプ / AssetCenterのバージョン情報]メニューを選択します。

「diff*.txt」ファイルの使用方法

「diff*.txt」ファイルがDOSテキスト形式 (またはASCII) であることを指定して、Excelまたは他のツールを使って開きます。

ヒント:

Excelを使用する場合、最初の行にオートフィルタを使うと、参照する変更事項ごとにフィルタを適用できるようになります。

列の内容は、最初の行に示されているタイトルで確認できます。

残りの行は、それぞれ標準データベースの構造の変更事項に対応します。

列の内容の例は、以下の通りです。

- オブジェクトを含むテーブルの名前
 - テーブルの作成

<4.2.0フォーマットの標準データベース内のSQL名>

- テーブルの削除

<旧フォーマットの標準データベースにおけるSQL名、または技術名>

- フィールド、インデックスまたはリンクの作成、削除または変更、テーブルの変更

<旧フォーマットの標準データベースにおけるSQL名、または技術名> (<4.2.0フォーマットの標準データベースにおけるSQL名>)

- 変更されたオブジェクト名
 - 削除されたオブジェクト

<旧フォーマットの標準データベースにおけるSQL名、または技術名>

- 変更されたオブジェクト

<旧フォーマットの標準データベースにおけるSQL名、または技術名> (<4.2.0フォーマットの標準データベースにおけるSQL名>)

- 追加されたオブジェクト

<4.2.0フォーマットの標準データベース内のSQL名>

- 説明
 - 変更または作成されたオブジェクト：オブジェクトの新しい説明
 - 削除されたオブジェクト：オブジェクトの古い説明

「diff*.htm」ファイルの使用方法

これらのファイルを参照するには、HTMLブラウザを使います。

「diff*.htm」ファイルの構造は以下の通りです。以下の太字の英語を探してファイル内を検索します。

1 Deleted table information

このタイトルは、削除されたテーブルを記述する各セクションの始めに挿入されています。

各テーブルにつき、以下の情報が含まれています。

- 削除されたテーブルに関する情報
- 削除されたテーブルのフィールド
- 削除されたテーブルのリンク
- 削除されたテーブルのインデックス

2 Inserted table information

このタイトルは、追加されたテーブルを記述する各セクションの始めに挿入されています。

各テーブルにつき、以下の情報が含まれています。

- 追加されたテーブルに関する情報
- 追加されたテーブルのフィールド
- 追加されたテーブルのリンク
- 追加されたテーブルのインデックス

3 Modified table

- Deleted objects
- Inserted objects
- Modified objects

「diff*.xml」ファイルの使用方法

「diff*.xml」ファイルは、XMLに精通している場合に必要に応じて使用すると便利です。

ユーザ企業側で「diff*.xml」ファイルを参照して、用途を決定してください。

手動で変換するアプリケーションデータ

この節には、変換時に検証するべきアプリケーションデータが列挙されていません。

ヒント:

これらのアプリケーションデータは、バージョン4.2.0で削除または変更された可能性のあるテーブル、フィールドまたはリンクを参照するため、確認する必要があります。

AssetCenter Script Analyzerを使って変更できるアプリケーションデータ

これらのアプリケーションデータは以下のカテゴリに分類されます。

- Basicスクリプト
- AQLクエリ
- テーブル名を格納するフィールド
- フィールド名を格納するフィールド
- ウィザード
- 特殊文字列（コンテキスト内でのリンクとフィールドの連結）

これらのデータやパラメータを確認する方法に関しては、本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション（シミュレーション用データベース）](#)」の章、「[手動で変換するアプリケーションデータを処理する](#)」の節を参照してください。

AssetCenterのグラフィカルインタフェースで、アプリケーションデータにアクセスできます。

変換時にこれらのデータは変更されません。

AssetCenter Script Analyzerは潜在的な問題を分析し、4.2.0データベース構造に適応するようにアプリケーションデータを変更します。

表 9.1. 手動変換するアプリケーションデータ - 一覧

テーブル (SQL 名)	フィールドまたはリンク (SQL名)	制約点
amAction	WizardScript	
	Script	
	MsgTo	
	MsgCc	
	MsgBcc	
	Subject	
	memMsgText	
	ActionFile	
	Folder	
	Parameters	
	DDEService	
	DDETopic	
	DDECommand	
	ContextTable	
	RefObject	
	amQuery	memQueryText
TableName		
amWfActivity	memScript	
	ContextTable	
amWfTransition	AQLCond	
	LinkToTargetCtxTbl	
	TargetContextTable	
amWfEvent	AQLCond	
	memScript	
	ContextTable	
	MonitTable	
	LinkToMonitTable	
amCalcField	memScript	

	Aql	バージョン4.0.0以降
	ComputeString	バージョン4.0.0以降
	TableName	
	Script	バージョン4.0.0以降
テーブル (SQL 名)	フィールドまたはリンク (SQL名)	制約点
	amAccessRestr	
	WriteCond	
	TableName	
	ReadCond	
	amTaxFormula	
	memFormula	
	TableName	
	amWfOrgRole	
	memScript	
	ContextTable	
	amFeatParam	
	AvailScript	
	DefValScript	
	MandatScript	
	ForceDspScript	
	HistScript	
	TableName	
	LinkFilter	
	amFeatScript	
	memScript	
	amOption	
	memOptValue	
	amFieldAdjustTempl	
	memScript	
	ContextTable	
	TargetField	
	amFieldAdjust	
	TargetField	
	AdjustedTable	バージョン4.0.0以降
	amDeprScheme	
	memScript	
	amLoan	
	ProrateField	
	amCntrRent	

テーブル (SQL名)	フィールドまたはリンク (SQL名)	制約点
	ProrateField	
	Description	バージョン4.6.0以降
amDateAlarm		
	MonitoredField	
	MonitoredTable	
amLabelRule		
	memScript	バージョン3.1.0、4.0.0とそれ以降
	TableName	バージョン3.1.0、4.0.0とそれ以降
	FieldName	バージョン3.1.0、4.0.0とそれ以降
amCatRefScript		
	memScript	バージョン4.0.0以降
amScriptLibrary		
	memScript	バージョン4.0.0以降
amCbkStoredEvent		
	FieldName	バージョン4.0.0以降
	Context	バージョン4.0.0以降
amCbkRule		
	AmountField	バージョン4.0.0以降
	EvtField	バージョン4.0.0以降
	Context	バージョン4.0.0以降
amCbkScript		
	Context	バージョン4.0.0以降
	memScript	バージョン4.0.0以降

確認するその他のアプリケーションデータ

以下のアプリケーションデータはAssetCenter Script Analyzerによって変換または検証されません。

- フィールドのヘルプ
- 書式
- ビュー
- インポートスクリプト
- 以下のWebページ
 - AssetCenter Web
 - Get-It
 - Get-Resources

- Connect-Itのシナリオ
- 出力スクリプトAssetCenter Export
- Crystal Reportsレポート

このため、これらのアプリケーションデータを1つずつテストする必要があります。

旧フォーマットの本番データベースの伝達された構造パラメータ

これらのパラメータはAssetCenter Database Administratorで定義されています。

旧フォーマットの本番データベースの構造パラメータは、「標準4.2.0gbase.dbb」ファイルに、構造変更の伝達の段階で伝達されます。

この手順の詳細については、本マニュアルの「[段階を追ってマイグレーションを実行する - シミュレーション \(シミュレーション用データベース\)](#)」の章、「[手順3-旧フォーマットの本番データベースの構造の変更事項を伝達する](#)」の節を参照してください。

表 9.2. 旧フォーマットの本番データベースの構造パラメータ - 一覧

データベースの オブジェクト	パラメータ	以下のバージョンで使用可能		
		3.0.1、3.0.2、 3.1.0、3.5.1、 3.5.2と3.6.0	4.0.0	4.1.0
テーブル	ラベル	はい	はい	はい
テーブル	説明	はい	はい	はい
テーブル	階層	いいえ	いいえ	はい
テーブル	任意管理項目を 含むことができ る	いいえ	はい	はい
テーブル	文字列	はい	はい	はい
テーブル	有効性	はい	はい	はい
テーブル	妥当性	いいえ	はい	はい
フィールドまた はリンク	ラベル	はい	はい	はい
フィールドまた はリンク	説明	はい	はい	はい
フィールド	サイズ	はい	はい	はい
フィールド	アップグレード	いいえ	はい	はい

データベースの オブジェクト	パラメータ	以下のバージョンで使用可能		
		3.0.1、3.0.2、 3.1.0、3.5.1、 3.5.2と3.6.0	4.0.0	4.1.0
フィールドまたはリンク	履歴保持	はい	はい	はい
フィールドまたはリンク	読取り専用	はい	はい	はい
フィールドまたはリンク	必須	はい	はい	はい
フィールドまたはリンク	無関連	いいえ	はい	はい
フィールド	フォーマット	はい	はい	はい
フィールドまたはリンク	デフォルト値	はい	はい	はい
フィールドまたはリンク	複製時にデフォルト値を再適用	はい	はい	はい

その他のマニュアル

マニュアル『AssetCenter 4.2.0 - マイグレーション』には、マイグレーションに直接関連する情報のみが記載されています。

本マニュアルに加えて、以下のマニュアルを参照することをお勧めします。

表 9.3. その他のマニュアル - 一覧

マニュアル名	内容	フォーマット	AssetCenterのインストール先フォルダにおけるパス
バージョン3.xと4.2.0の相違点	<ul style="list-style-type: none"> バージョン4.2.0の新機能 	印刷版 オンライン	/doc/pdf/diff*.pdf /doc/pdf/diff*.pdf
Readme	<ul style="list-style-type: none"> 最新情報 	テキスト	readme.txt
リリースノート	<ul style="list-style-type: none"> AssetCenterの付属マニュアルのリスト 新機能の概要 	印刷版 オンライン	/doc/pdf/relnotes*.pdf /doc/pdf/relnotes*.pdf

マニュアル名	内容	フォーマット	AssetCenterのインストール先フォルダにおけるパス
インストール	<ul style="list-style-type: none"> AssetCenterプログラムのリスト サポートされているオペレーティングシステムと必要最小限の動作環境 サポートされるDBMS AssetCenterのインストール 	印刷版 オンライン	<code>/doc/pdf/installbook*.pdf</code> <code>/doc/chm/installbook*.chm</code>
データベースの構造	<ul style="list-style-type: none"> データベースのテーブル、フィールド、リンク、インデックスの一覧 AssetCenterが自動的にトリガするエージェント 	テキスト ファイル 印刷版 オンライン	<ul style="list-style-type: none"> <code>doc/infos/database.txt</code> <code>doc/infos/tables.txt</code> <code>/doc/pdf/dbstruct*.pdf</code> <code>/doc/chm/dbstruct*.chm</code>
異なるバージョン間の構造の相違点	<ul style="list-style-type: none"> 変更されたテーブル、フィールド、リンク、インデックスの一覧 	テキスト Html	<code>doc/infos/diff*.txt</code> <code>/infos/diff*.html</code>
管理	<ul style="list-style-type: none"> AssetCenter Database Administrator インポート 	印刷版 オンライン	<code>/doc/pdf/admin*.pdf</code> <code>/doc/chm/admin*.chm</code>
AssetCenterの高度な使い方	<ul style="list-style-type: none"> データの出力 	印刷版 オンライン	<code>/doc/pdf/advan*.pdf</code> <code>/doc/chm/advan*.chm</code>

XMLの詳細については、Webサイト<http://www.w3.org/XML/>を参照してください。



October 10, 2002