

**Sterling Selling and Fulfillment Foundation**



## **構成・デプロイ・ツール・ガイド**

*バージョン 9.1*



**Sterling Selling and Fulfillment Foundation**



## **構成・デプロイ・ツール・ガイド**

*バージョン 9.1*

**お願い**

本書および本書で紹介する製品をご使用になる前に、67ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM Sterling Selling and Fulfillment Foundation バージョン 9.1、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

**原典：** Sterling Selling and Fulfillment Foundation  
Configuration Deployment Tool Guide  
Version 9.1

**発行：** 日本アイ・ビー・エム株式会社

**担当：** トランスレーション・サービス・センター

第1刷 2012.2

© Copyright IBM Corporation 1999, 2011.

# 目次

<b>第 1 章 デプロイメント・ツールの概要</b> . . . . .	<b>1</b>
デプロイメント・ツールの概要 . . . . .	1
<b>第 2 章 CDT の概念</b> . . . . .	<b>3</b>
CDT のソースおよびターゲット環境 . . . . .	3
単一スキーマおよびマルチスキーマのデプロイメン トに関するガイドライン . . . . .	3
CDT の構成グループおよびドライバー・エンティテ ィー . . . . .	7
外部で管理されている構成データのインポート . . . . .	7
CDT を使用してカスタム・テーブルをデプロイする 方法 . . . . .	8
CDT が外部キー検査を処理する方法 . . . . .	9
CDT を使用したデータの変換 . . . . .	9
CDT リソース・ファイル . . . . .	10
<b>第 3 章 CDT インターフェースについて</b> . . . . .	<b>11</b>
CDT インターフェースについて . . . . .	11
CDT の「比較結果 (Comparison Results)」ウィンド ウ . . . . .	12
デプロイメント・ツールの「ステータス」パネル . . . . .	13
<b>第 4 章 構成デプロイメント・ツールを使 用する前に</b> . . . . .	<b>15</b>
CDT のシステム要件 . . . . .	15
CDT セキュリティー戦略 . . . . .	16
CDT の変更管理戦略 . . . . .	16
CDT ロールバック戦略 . . . . .	17
CDT およびアップグレードとメンテナンスについて アップグレード・シナリオの例 . . . . .	17
<b>第 5 章 GUI モードでの CDT のセット アップおよび使用</b> . . . . .	<b>19</b>
GUI モードでの CDT のセットアップおよび使用 . . . . .	19
GUI モードの CDT コマンド・ライン引数 . . . . .	19
GUI モードでの CDT の始動 . . . . .	19
GUI モードでの CDT の停止 . . . . .	23
GUI モードでの CDT 設定の実行 . . . . .	23
<b>第 6 章 テキスト・モードでの CDT の セットアップおよび使用</b> . . . . .	<b>27</b>
テキスト・モードでの CDT のセットアップおよび 使用 . . . . .	27
テキスト・モードの CDT コマンド・ライン引数 . . . . .	27
テキスト・モードでの CDT の実行 . . . . .	28
テキスト・モードでの CDT 設定の実行 . . . . .	29
<b>第 7 章 CDT を使用したエレメントの変 換について</b> . . . . .	<b>33</b>
CDT を使用したエレメントの変換について . . . . .	33

CDT を使用したエレメントの変換 . . . . .	34
-----------------------------	----

<b>第 8 章 CDT のソース・データベースと ターゲット・データベースの比較</b> . . . . .	<b>35</b>
CDT のソース・データベースとターゲット・デー タベースの比較 . . . . .	35
CDT を使用したソース・データとターゲット・デー タの比較 . . . . .	35
CDT を使用した比較対象組織の指定 . . . . .	37
CDT を使用したマルチスキーマ環境でのデータ の比較について . . . . .	38
CDT を使用したデータベースの差異の検証 . . . . .	38
CDT を使用した比較結果のエクスポート . . . . .	40
CDT を使用した差異のレポートの生成 . . . . .	41
CDT を使用した構成の差異のインポート . . . . .	42
<b>第 9 章 構成データのデプロイ</b> . . . . .	<b>45</b>
構成データのデプロイ . . . . .	45
CDT を使用した構成データのデプロイ . . . . .	45
CDT によって生成される監査 . . . . .	46
CDT によって使用されるドライバー XML ファ イル . . . . .	46
コマンド・ライン・モードでの構成データのデ プロイ . . . . .	46
<b>第 10 章 構成データ・バージョン管理ツ ールの概要</b> . . . . .	<b>49</b>
構成データ・バージョン管理ツールの概要 . . . . .	49
構成データ・バージョン・ラベルの例 . . . . .	50
CDV ツール・プリファレンスの設定について . . . . .	51
構成データ・バージョン管理ツールのプリファ レンスの設定 . . . . .	52
CDV ツールの実行 . . . . .	53
構成データ・バージョン管理ツールからの結果の エクスポート . . . . .	57
構成データ・バージョン管理ツールを使用した データベースの競合の解決 . . . . .	58
さまざまなタイプの競合を解決する方法 . . . . .	58
CDV ツールによって生成されるバージョン監査 ファイルのインポート . . . . .	59
CDV ツールを使用したデータベース変更のデ プロイ . . . . .	60
CDV ツールを使用した変更のロールバック . . . . .	61
<b>第 11 章 構成デプロイメント・ツールの トラブルシューティング</b> . . . . .	<b>63</b>
CDT のトラブルシューティング . . . . .	63
構成デプロイメント・ツールのメッセージ . . . . .	63
構成デプロイメント・ツールでの予期しないエラ ー・メッセージ . . . . .	65

cdtshell スクリプトで構成データをエクスポートする  
場合の例外 . . . . . 65  
データ・ロールバック・スクリプト . . . . . 66  
**特記事項 . . . . . 67**

**索引 . . . . . 71**

---

# 第 1 章 デプロイメント・ツールの概要

---

## デプロイメント・ツールの概要

IBM® Sterling Selling and Fulfillment Foundation の増分構成中に、一般的に変更はテスト環境で作成され、実稼働環境にロールアウトされます。構成データのマイグレーションは、かなり面倒で時間がかかる作業になる可能性があります。 Sterling Selling and Fulfillment Foundation は、構成データをマイグレーションできるようにする構成デプロイメント・ツールを提供しています。このツールは、データ転送時のシステムのダウン時間を削減し、精度を確保するために求められる労力を最小限に抑えながら、データ保全性を確保します。

このツールは、標準の日常操作の一環として変更されるデータのマイグレーションを目的として設計されています。構成デプロイメント・ツールはアップグレードの結果である構成データをデプロイするために使用できますが、データ・アップグレード自体の実行には使用すべきでないことに注意してください。

構成デプロイメント・ツール (CDT) には、Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench (「WorkBench」として知られる) からアクセスできます。

UNIX サーバー上の Sterling Selling and Fulfillment Foundation 構成デプロイメントツールの UI を Windows クライアントから実行するには、UNIX サーバーに XWindows がインストールされている必要があります。

CDT は、以下の機能を提供します。

- 構成データ全体または一部あるいは個別の論理部分を転送する。
- データを XML ファイルまたはデータベースとの間で転送する。
- ネットワーク構成に応じて、2 つの環境で異なる、IP アドレスやポート番号のような特定のデータを変換する。
- 2 つのシステムを比較することにより、構成の差異のレポートを生成する。





---

## 第 2 章 CDT の概念

---

### CDT のソースおよびターゲット環境

Sterling Selling and Fulfillment Foundation 構成デプロイメント・ツールは、ある Sterling Selling and Fulfillment Foundation 環境から別の環境にデータをデプロイします。デプロイメントは、開発環境からテスト環境へ、ステージング環境から実稼働環境へ、またはその他の組み合わせで行われる可能性があります。データのデプロイ元として機能する環境は「ソース」環境として知られています。データのデプロイ先として機能する宛先環境は「ターゲット」環境として定義されます。このデプロイメントは、データベースまたは XML ファイルとの間でのデータのインポートおよびエクスポートという形態を取ることができます。

ソース・データベースの構成データを変更すると、CDT の既存のトランザクションが影響を受ける場合があります。

また、YFS\_TRAN\_LOCN\_ATTRS テーブルの場合、構成データのみターゲット・データベースにコピーされます。Pend\_in\_volume、Pend\_in\_weight、および Freeze on variance などのトランザクション・データ列は、ターゲット・データベースにコピーされません。重量や容積は再計算され、ターゲット・データベースで更新されます。

以下の例は、これらの XML ファイルの必須フォーマットを示しています。

```
<?xml version="1.0" encoding="UTF-8" ?>
<YFS_ATP_RULESList TableName="YFS_ATP_RULES">
  <AtpRules AccumulationDays="730" AdvanceNotificationTime="0"
    AtpRule="DEFAULT" AtpRuleKey="DEFAULT" AtpRuleName="DEFAULT"
    BackwardConsumptionDays="730" ConsiderPoForAlloc="N" Createprogid=""
    Createts="" Createuserid="SYSTEM" ForwardConsumptionDays="730" Lockid="0"
    MaxInventoryHoldDays="730" Modifyprogid="SYSTEM" Modifys=""
    Modifyuserid="SYSTEM" OrganizationCode="DEFAULT" PastDueDemandDays="730"
    PastDueSupplyDays="730" ProcessingTime="0" />
</YFS_ATP_RULESList>
```

図 1. 構成データ XML

### 単一スキーマおよびマルチスキーマのデプロイメントに関するガイドライン

CDT を使用して、単一スキーマおよびマルチスキーマ環境でデータを比較およびデプロイできます。どちらの環境でも、データは以下の構成グループに編成されます。

- メタデータ - 中核となる構成情報のデータベース検索に必須です。これは、さまざまなバージョンにわたる固有のテーブルのセットで、マルチスキーマ、マルチバージョンのデプロイメント間での接続の「トラフィックの方向付け」を行います。
- 構成データ - 以下のグループで構成されます。

- HUB データ - HUB に関するデータおよびルールが含まれます。
- 組織によって決定される構成データ - 一般に、システムまたはビジネス・ルール (ソーシング・ルール、ルーティング・ガイド、出荷プリファレンスなど)、および組織を保管します。
- マスター・データ - 次のグループで構成されます。
  - 組織によって決定されるマスター・データ - バッチ・フィードによって作成され、他の組織およびユーザーによって頻繁に参照されるデータが含まれます。
  - 出荷ノードによって決定されるマスター・データ - 出荷ノードに関連するマスター・データが含まれます。

構成グループの詳細については、7 ページの『CDT の構成グループおよびドライバー・エンティティ』を参照してください。

## CDT を使用した単一スキーマ環境でのデータのデプロイ このタスクについて

単一スキーマ環境では、データを比較し、組織別にデプロイできます。

組織別にデータをデプロイするには、以下の手順に従います。

### 手順

1. DEFAULT 組織、および DEFAULT 組織に参加する組織の構成データ構成グループをデプロイします。
2. 他の組織によって参照されている構成データ (構成の継承、運送会社、カタログ/価格設定組織など) を保持する組織の構成データ構成グループをデプロイします。
3. マスター・データ構成グループをデプロイします。

### タスクの結果

データのデプロイについて詳しくは、19 ページの『GUI モードでの CDT の始動』を参照してください。

## CDT の単一スキーマのデプロイメントの例

以下は、例 1 および 2 について構成された参加者の設定です。

- DEFAULT
- ORG1
- ORG2 - ORG1 から構成を継承
- ORG3 - 独自の構成を定義
- CARRIER1 - DEFAULT に参加する運送会社
- CARRIER2 - ORG2 および ORG3 に参加する運送会社

### 単一スキーマのデプロイメント例 1: このタスクについて

ファクトリー設定で ORG2 をソースからターゲットにデプロイするには、以下のようになります。

## 手順

1. DEFAULT 組織および CARRIER1 の構成データをデプロイします。
2. ORG1、ORG2、および CARRIER2 の構成データをデプロイします。

CARRIER2 は ORG2 に参加します。このため、CARRIER2 は ORG2 でデプロイする必要があります。

3. ORG2 のマスター・データをデプロイします。

## 単一スキーマのデプロイメント例 2:

### このタスクについて

ファクトリー設定で ORG3 をソースからターゲットにデプロイするには、以下のようになります。

## 手順

1. DEFAULT および CARRIER1 の構成データをデプロイします。
2. ORG3 および CARRIER2 の構成データをデプロイします。
3. ORG3 のマスター・データをデプロイします。

## CDT を使用したマルチスキーマ環境でのデータのデプロイ

### このタスクについて

マルチスキーマ環境では、データを比較し、組織またはコロニー別にデプロイできます。メタデータ構成グループ、組織データ構成グループ、およびマスター構成グループを、対応するスキーマと照らし合わせて、個別にデプロイする必要があります。一般的なマルチスキーマ構成について詳しくは、「*Sterling Selling and Fulfillment Foundation マルチ・テナント・エンタープライズ・ガイド*」を参照してください。

組織またはコロニー別にデータをデプロイするには、以下の手順に従います。

## 手順

1. メタデータ構成グループをデプロイします。この手順は、テスト環境から実稼働環境に移行する場合のみ必要です。
2. DEFAULT 組織、および DEFAULT 組織に参加する組織の構成データ構成グループをデプロイします。
3. 他の組織によって参照されている構成データ (構成の継承、運送会社、カタログ/価格設定組織など) を保持する組織の構成データ構成グループをデプロイします。
4. 他のコロニーまたは他のコロニーに属する組織の構成データ構成グループをデプロイします。
5. マスター・データ構成グループをデプロイします。
6. すべてのデータをデプロイした後、「*Sterling Selling and Fulfillment Foundation 分散オーダー管理 構成ガイド*」の「時間トリガー・トランザクション参照 (Time-Triggered Transaction Reference)」という付録の説明に従って、コロニー・マップ・シンクロナイザー・エージェントを実行します。

## CDT マルチスキーマのデプロイメントの例

以下は、例 1 および 2 について構成された参加者の設定です。

- DEFAULT COLONY
  - DEFAULT
  - ORG1
  - ORG2 - ORG1 から構成を継承
  - ORG3 - 独自の構成を定義
  - CARRIER1 - DEFAULT に参加
- COLONY1
  - ORG4
  - ORG5
  - CARRIER2 - ORG4 に参加

### マルチスキーマのデプロイメント例 1:

#### このタスクについて

ファクトリー設定で ORG2 をソースからターゲットにデプロイするには、以下のようになります。

#### 手順

1. DEFAULT 組織および CARRIER1 の構成データをデプロイします。
2. ORG1 および ORG2 の構成データをデプロイします。
3. ORG2 のマスター・データをデプロイします。
4. コロニー・マップ・シンクロナイザーを実行します。

### マルチスキーマのデプロイメント例 2:

#### このタスクについて

ファクトリー設定で COLONY1 をソースからターゲットにデプロイするには、以下のようになります。

#### 手順

1. DEFAULT 組織および CARRIER1 の構成データをデプロイします。
2. COLONY1 の構成データをデプロイします。COLONY1 には、ORG4、ORG5、および CARRIER2 の構成データが含まれます。

CARRIER2 は ORG4 に参加します。このため、CARRIER2 は ORG4 でデプロイする必要があります。

3. COLONY1 のマスター・データをデプロイします。
4. コロニー・マップ・シンクロナイザーを実行します。

---

## CDT の構成グループおよびドライバー・エンティティ

Sterling Selling and Fulfillment Foundation 構成データ全体は、「構成グループ」および「ドライバー・エンティティ」と呼ばれる論理サブセットに分類されます。構成グループおよびドライバー・エンティティは事前定義されており、変更できません。

デプロイメント・プロセス中に、ターゲット・データベースとソース・データベースが一致するように、より細分化された挿入、更新、および削除を実行する必要がある場合、これらの構成グループまたはドライバー・エンティティを選択します。

### ドライバー・エンティティ

論理エンティティ (組織やパイプラインなど) をはじめとして、ほとんどの Sterling Selling and Fulfillment Foundation 構成データはデプロイできます。これらの論理エンティティは「ドライバー・エンティティ」と呼ばれます。ドライバー・エンティティは、データ保全性を損なうことなくソースからターゲットにデプロイできる、最も細分性レベルの高い情報を表します。

レコード・レベルでのデプロイメントが可能なのは、ドライバー・エンティティのみです。その他のテーブルについては、以下の条件のいずれかが適用されます。

- テーブルがどのドライバー・エンティティにも依存していない場合、テーブルは完全にデプロイされます。
- ドライバー・エンティティに対応するレコードのみデプロイされます。

ドライバー・エンティティに関する情報は複数のテーブルに保管できます。また、エンティティのデプロイ時には、データの整合性を保持するために、関連するすべてのテーブルに含まれるデータが 1 つのトランザクション境界にまとめられてデプロイされます。

### 構成グループ

また、論理的に関連したテーブルまたはドライバー・エンティティも「構成グループ」にまとめられます。これは一般的に、Sterling Selling and Fulfillment Foundation 内の、より大きく重要な論理データ・モデルを表します。例として、ビジネス・プロセス・モデルまたは参加者モデルが挙げられます。これらのグループは、ユーザー・インターフェース上でのナビゲーションの利便性と容易性を図るために提供されています。

---

## 外部で管理されている構成データのインポート

Sterling Selling and Fulfillment Foundation の実装環境では、ソース Sterling Selling and Fulfillment Foundation データベースの一部ではない特定のデータをターゲットにインポートしなければならない場合があります。こうしたテーブルの場合、CDT は正しいデータにアクセスできないため、CDT を使用してデータをデプロイしてはなりません。

## ベスト・プラクティス

構成デプロイメント・ツールを使用して外部で管理されているデータをデプロイする必要がある場合、これを処理する方法として推奨されるのは、このデータをソースにインポートしてから、CDT を使用してターゲットにデプロイすることです。これにより、データ保全性が保証されます。

このデータをソース・データベースにインポートできない場合、Sterling Selling and Fulfillment Foundation は、ターゲット・データベースが確実にこれらのテーブルを無視するか、追加するようにすることで、外部データの処理を可能にする機能を提供します。無視および追加専用機能の使用は、使用可能な他のすべてのオプションを試行し、かつ環境に厳密なテストを課した後のみに限定します。

### 注意:

無視または追加専用機能を使用している場合、CDT は外部データの保全性を保証できません。データ保全性を確保するには、CDT が構成データに完全にアクセスできなければなりません。

### 無視

テーブル内のデータが外部から保守される場合、それらのテーブルを無視するようにプリファレンスを指定することにより、デプロイメント操作でそれらのテーブルを除外できます。

テーブルまたはドライバー・エンティティーを無視すると、従属テーブルもすべて自動的に無視されます。ただし、複数のドライバー・エンティティーのデータを保管し、複数のグループに存在するテーブルがあります。その例として、パイプライン、サービス、およびステータスのデータを含む YFS\_GRAPH\_UI テーブルがあります。これらのテーブルのいずれかを無視すると、CDT は対応するレコードに誤って削除のマークを付けます。

### 追加専用モード

テーブルが部分的に外部から保守される場合、これらのテーブルが「追加専用」モードでデプロイされるようにプリファレンスを指定できます。

追加専用テーブルの場合、従属テーブルは無視されません。テーブルに追加専用としてマークを付けると、ターゲット・データベース内の数行のみがソース・システムで保守され、それ以外の行は外部的にインポートされることを意味します。そのような場合、ソース・システムに存在するデータと外部システムに存在するデータがオーバーラップしないようにすることが非常に重要です。例えば、出荷ノードをソース・データベースで保守し、ストア情報をターゲットに直接インポートする場合、ソース・データベースにストアが含まれていてはなりません。含まれていると、予測できない結果が生じます。

---

## CDT を使用してカスタム・テーブルをデプロイする方法

CDT は、Sterling Selling and Fulfillment Foundation データベース・フレームワーク内で定義された構成テーブルおよび拡張テーブルを自動的にデプロイします。カスタム (Sterling Selling and Fulfillment Foundation 以外の) 構成テーブルがインストール済み環境で定義されている場合、CDT を、それらのテーブルをデプロイするため

の構成にする必要があります。 CDT がそれらのテーブルをデプロイできるようにするには、 `cdt_custom.xml` という特殊なカスタム・デプロイメント XML ファイルを作成することにより、テーブルを CDT に登録する必要があります。このファイルのサンプルが、 `<INSTALL_DIR>/resources/ydkresources` ディレクトリにあります。このファイルは、「Custom Tables」という名前のグループを定義します。これには、カスタム・テーブルのリストが含まれていなければなりません。 CDT は、1 つ以上の 1 次キー列を持つすべてのテーブルについて、カスタム・レコードに対する変更を自動的に比較し、表示して、デプロイします。

このツールは、ドライバーとしてのカスタム・テーブルまたは従属ツリー構造でのカスタム・テーブルの表示をサポートしていません。そのため、すべてのカスタム・テーブルをまとめてデプロイできるのは、「Custom Tables」グループの一部としてデプロイする場合に限られます。また、このツールは、1 次キーのないカスタム・テーブルもサポートしていません。

`cdt_custom.xml` ファイルには、以下が含まれます。

```
<Group Name="Custom Tables">
  <Table Name="CUSTOM_CONFIG_TABLE_1"/>
  <Table Name="CUSTOM_CONFIG_TABLE_2"/>
</Group>
```

---

## CDT が外部キー検査を処理する方法

CDT は、エンティティを定義するすべての関連テーブルを一度の操作でまとめてデプロイすることにより、データ整合性を維持します。さらに、データ保全性を確保するために、CDT はテーブルごとに必要な外部キー制約も検査しますが、これは、全く異なるグループのテーブルに対して定義されている可能性があります。そのため、データの小さいサブセットをデプロイする際に、従属テーブル内の対応するデータが同じ操作でデプロイされていない場合には、外部キー制約違反を示すエラー・メッセージが表示される可能性があります。この場合、より大きいデータ・セットでデプロイしてみてください。外部キー制約は、カスタム・テーブルに対しては定義または検査されないことに注意してください。

最高のパフォーマンスを実現するために、Sterling Selling and Fulfillment Foundation 構成全体をデプロイする際には、外部キー制約は検査されません。

---

## CDT を使用したデータの変換

開発環境と実稼働環境で、サーバー名や IP アドレスなどのネットワーク設定の値が異なることはよくあります。 Sterling Selling and Fulfillment Foundation の構成データ・テーブルの中には、ホスト名、IP アドレス、および URL を保管するものがあります。これらはソース環境では有効ですが、このデータをターゲット環境にデプロイする場合、ターゲット環境に適用可能な、対応する値で構成を更新する必要があります。 CDT は、データをターゲットにデプロイする前に ソース・データに対して実行される変換を指定することにより、これらのデータ・エレメントをターゲットに適した値に自動的に変換できるようにします。

---

## CDT リソース・ファイル

CDT 固有のリソース・ファイルは、<INSTALL\_DIR>/resources/ydkresources ディレクトリにあります。これらのリソース・ファイルを使用して、CDT 設定、データベース固有の設定などを変更できます。

### **cdt\_custom.xml.sample**

cdt\_custom.xml.sample は、インストール環境で定義されたカスタム・テーブルをデプロイするために使用できる、特殊なカスタム・デプロイメント XML ファイルのサンプルです。カスタム・テーブルのデプロイについて詳しくは、8 ページの『CDT を使用してカスタム・テーブルをデプロイする方法』を参照してください。

### **cdt\_dbdefaults.properties**

この cdt\_dbdefaults.properties ファイルは、所定のデータベース・タイプに対して追加のデータベース・プール・オプションを指定するために使用されます。このファイルには、CDT の構成中に使用される各種データベースのデフォルト値が含まれます。CDT は、このファイルを使用して、データベース接続を処理します。

### **ydkprefs.xml**

ydkprefs.xml ファイルには、CDT に対して定義された設定が含まれており、GUI ベースの CDT を実行する際に作成されます。これには、CDT の比較操作の動作を決定する構成プリファレンス (レポート・ディレクトリーなど) およびパラメーターが含まれます。

### **ydkprefs.xml.sample**

ydkprefs.xml.sample ファイルは、Sterling Selling and Fulfillment Foundation のインストール環境に自動的に組み込まれます。このファイルを編集して、CDT の比較操作の動作を決定する構成プリファレンス (レポート・ディレクトリーなど) およびパラメーターを指定できます。このサンプル・ファイルの変更が終了したら、ydkprefs.xml に名前変更します。



---

## 第 3 章 CDT インターフェースについて

---

### CDT インターフェースについて

構成デプロイメント・ツールを始動すると、セッション中に使用するソース・データベースおよびターゲット・データベースの詳細を指定するためのプロンプトが出されます。ソース・データベースおよびターゲット・データベースへの接続が成功すると、「Deployment Explorer」ウィンドウが表示されます。

#### CDT Deployment Explorer

「Deployment Explorer」ウィンドウには、構成グループ、ドライバー・エンティティ、およびデプロイできるテーブルのリストが表示されます。ソース・データベースおよびターゲット・データベースに定義された名前が、見出しパネルに表示されます。

構成デプロイメント・ツールにログインするたびに、このウィンドウの 1 つのインスタンスが表示されます。

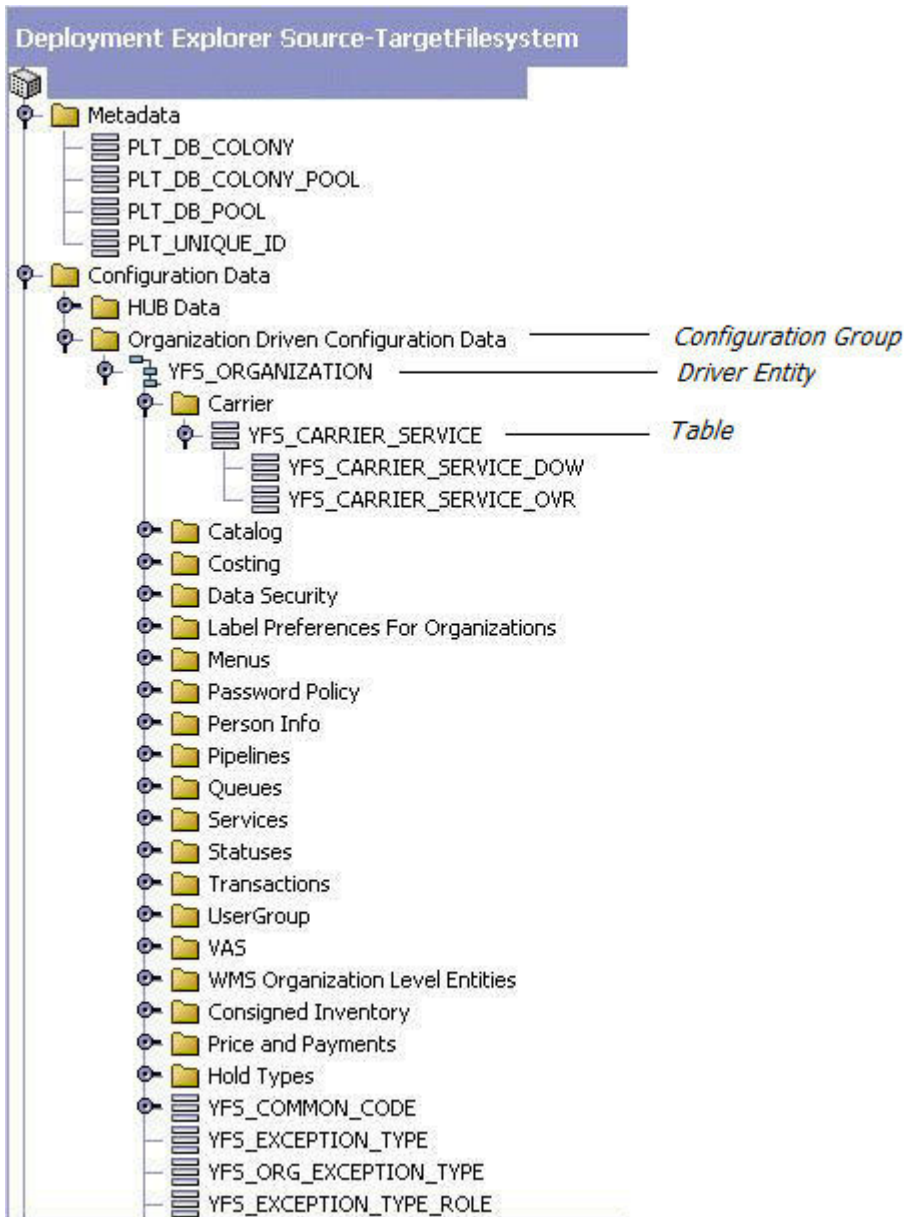


図 2. Deployment Explorer

ソース・データベースとターゲット・データベースの間で比較する構成グループまたはドライバー・エンティティを選択できます。

比較操作中に、比較操作の進捗および結果が「比較結果 (Comparison Results)」ウィンドウおよび「ステータス」パネルに表示されます。

## CDT の「比較結果 (Comparison Results)」ウィンドウ

「比較結果 (Comparison Results)」ウィンドウには、ソース・データベースとターゲット・データベースを比較した結果が表示されます。

「比較結果 (Comparison Results)」ウィンドウには、現行セッションに関する情報が表示されます。各セッション中に表示できる「比較結果 (Comparison Results)」ウィンドウは、1 つのみです。ある比較の結果を表示した後、別のテーブルのセットを比較できるようにするには、ウィンドウを閉じる必要があります。

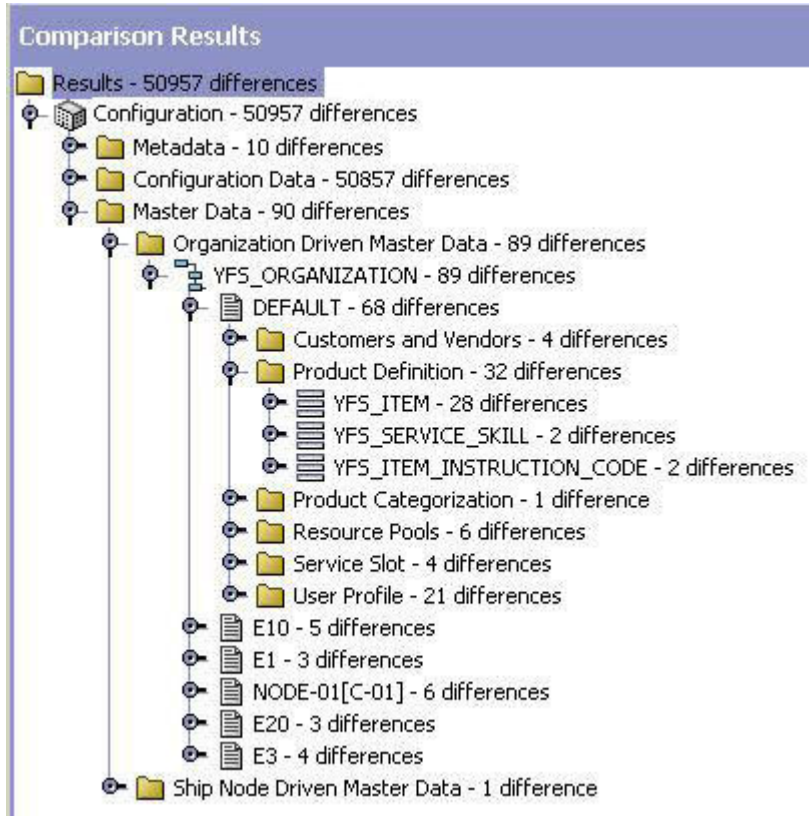


図3. 「比較結果 (Comparison Results)」ウィンドウ

比較結果を生成した後、以下のいずれかのタスクを実行できます。

- 差異のレポートの生成
- 各差異の詳細の表示
- ソース・データベースからターゲット・データベースへの構成データのデプロイ

---

## デプロイメント・ツールの「ステータス」パネル

「ステータス」パネルには、実行中の操作に関する情報が表示されます。

```
Analyzing XXXXXXXXXX 00:05:30
12:27:58->Reading YFS_REGION_LEVEL
12:27:59->Reading YFS_REGION_LEVEL_ASSOC
12:27:59->Reading YFS_REGION_SCHEMA
12:28:00->Reading YFS_REGION
12:28:00->Reading YFS_REGION_DETAIL
12:28:00->Reading YFS_REGION_BEST_MATCH
12:28:01->Reading PLT_QUALIFIED_TAG_TYPE
12:28:01->Reading PLT_QUALIFIED_TAG
12:28:02->Reading PLT_QUALIFIED_TAG_VER_COMPTBLT
12:28:02->Reading YFS_ORDER_TAG_DETMIN
12:28:02->Reading YFS_ORDER_TAG_MOD_TYPE
12:28:03->Reading YFS_CONDITION
12:28:04->Reading YFS_CONDITION_PROPS
12:28:04->Reading YFS_RESOURCE
```

図4. 構成デプロイメント・ツールの「ステータス」パネル

## 第 4 章 構成デプロイメント・ツールを使用する前に

### CDT のシステム要件

構成デプロイメント・ツールの RAM 要件は、データベースのサイズおよび構成データの分布によって異なります。

#### 時間の見積もり

構成デプロイメント・ツールが比較およびデプロイメント・タスクの実行に要する時間は、システム・リソースおよび構成データのサイズと分布に応じて異なります。例えば、あるテーブルに、他のテーブルの外部キー制約によって参照される多数のレコードが含まれる場合、またはドライバー・エンティティの機能をするテーブルに多数のレコードが含まれる場合、処理時間が増加することがあります。

マイグレーションするレコードの数は、デプロイメントごとに多くても 200,000 個までにするよう推奨されています。

構成デプロイメント・ツールのテスト中に、Sterling Selling and Fulfillment Foundation は、Pentium クラスのマシン (512 MB の RAM を搭載、550 MHz で稼働) でタスクを実行するために要する時間を測定します。構成デプロイメント・ツールは、表 1 で説明されているように実行されます。

表 1. 構成デプロイメント・ツールを使用した時間の見積もり

タスク	データベースの説明	時刻
比較	ソース・データベース - 110,000 レコード ターゲット・データベース - 110,000 レコード	7 分
比較	ソース・データベース - 110,000 レコード ターゲット・データベース - 空	4 分
デプロイメント	110,000 の差異	11 分

注: 3 GB の RAM を搭載し、2 GHz で稼働する 64 ビットのマシンで行われた構成デプロイメント・ツールのテストでは、CDT が 1 回当たり最大 750,000 個のレコードを比較し、デプロイできることを示しています。

#### インストール

構成デプロイメント・ツールは、Sterling Selling and Fulfillment Foundation のインストール時に自動的にインストールされます。

## 環境状態

構成デプロイメント・ツールは、ソース環境とターゲット環境が以下の点で完全に一致することを想定しています。

- Sterling Selling and Fulfillment Foundation のリリース (フィックスパックを含む)
- JDBC ドライバーのリリース
- データベース・ソフトウェアのリリース
- データベース構造 (テーブル、インデックス、シーケンスなどのスキーマ・オブジェクト)

注: 構成デプロイメント・ツールを使用する前に、ソース環境とターゲット環境の両方でファクトリー設定を必ず実行してください。

構成デプロイメント・ツールは技術専門家が不定期に実行する作業に使用するため、ローカライズやカスタマイズは**できません**。ただし、構成プリファレンスは指定できます。

---

## CDT セキュリティー戦略

構成デプロイメント・ツールは、データベース・プロバイダーによって提供されるユーザー認証および許可を利用します。アクセス制御および許可は、構成デプロイメント・ツールでは指定されません。

構成デプロイメント・ツールのユーザーが両方のデータベースに対して十分な認証特権 (選択、挿入、更新、および削除) を持っていることを確認してください。全 DBA 特権が必要というわけではありません。

---

## CDT の変更管理戦略

構成デプロイメント・ツールは、他の手段を使用したソースまたはターゲット・スキーマに対する構成データの変更を制限するための検査を実行**しません**。各自が独自の метод論を作成して適用する必要があります。

例えば、構成デプロイメント・ツールを使用して、データをステージング・データベースから実動データベースにマイグレーションする場合、実稼働環境の構成がこのツール以外の手段で変更されることは**予期されていません**。そのような場合、変更は CDT の次回実行時に上書きされます。また、CDT の実行中にソースまたはターゲットのいずれかで変更が行われると、データ保全性の問題が生じる可能性があります。

構成デプロイメント・ツールは、アプリケーション・マネージャーまたはその他の手段を使用して構成データが実稼働環境で直接変更される実装では、サポート**されません**。実稼働環境の構成データに対する緊急のフィックスまたは重大なフィックスといった例外的な場合には、ステージング・データベースに同じ変更を適用して更新する必要があります。

---

## CDT ロールバック戦略

アプリケーションの障害およびダウン時間を回避するには、厳密なロールバック方法論を実装します。この方法論には、構成デプロイメント・ツールを使用して変更をデプロイする前に、実稼働環境の構成データのバックアップ・スナップショットを作成することが含まれます。このバックアップは、データベース固有のエクスポートおよびインポート・ユーティリティーを使用して実行されます。 Sterling Selling and Fulfillment Foundation には、Oracle、DB2®、および Microsoft SQL Server データベースのサンプルが用意されています。これを各自の使用状況に合わせてカスタマイズできます。データ・ロールバックについて詳しくは、66 ページの『データ・ロールバック・スクリプト』を参照してください。

---

## CDT およびアップグレードとメンテナンスについて

構成デプロイメント・ツールの使用が、マルチステップ・ステージング環境でアップグレードまたはフィックスバックを適用するための方法論に影響を与えることがあってはなりません。

適用するアップグレード方法論を、実動前に既にステージング向けにセットアップされている環境用に変更してはなりません。しかし、構成デプロイメント・ツールが実行できるのは構成データのデプロイに限られるため、このツール自体は、マルチステップ・アプリケーションのステージングおよびデプロイメント環境のサポートに必要なすべてのプロセスおよび方法論のサポートを提供するわけではありませ

ん。

製品のアップグレードおよびフィックスバックの適用プロセスは、ステージング領域を実動領域と同期させておく必要がある環境では、特に複雑です。これらの環境の調和を保つための 1 つの方法は、ソフトウェア・フィックスバックを両方のシステムに同時に適用し、データ・アップグレードをリバース・デプロイすることで

す。これは、アプリケーションのデータ・アップグレードが異なる動作をし、扱うトランザクション・データに基づいて異なる結果を生む場合があるためです。このアプリケーションのデータ・アップグレードが実動領域とステージング領域で別々に実行されると、アップグレード・プログラムが扱ったトランザクション・データに差異が生じるため、結果は大きく異なる場合があります。そのような場合、実動スナップショットを基準と見なし、ステージング領域にリバース・デプロイする必要があります。このことは、実動データベースをソースとして構成し、ステージング・データベースをターゲットとして構成することによって行えます。

### アップグレード・シナリオの例

アップグレード・シナリオの例では、Sterling Selling and Fulfillment Foundation は、オーダー・タイプのさまざまな属性を認識する機能を導入します。例えば、Order\_type "URGENT" は、ユーザー・インターフェースにおいて、オーダー同士を区別できる固有のアイコンを使用してオーダーを表示しなければならないことを意味します。しかし、前のリリースでは、Order\_type フィールドを使用してオーダーを他のタイプに分類していました。このフィールドはオーダーの分類を目的としていたためです。

Sterling Selling and Fulfillment Foundation がアップグレード・ツールキットを提供している場合、ツールキットの 1 つのコンポーネントが **Order\_type** フィールドへのアップグレードを処理します。

アップグレード・ロジックの流れは以下のとおりです。

1. **Order\_type** フィールドのすべての個別値を YFS\_ORDER テーブルから読み取りません。
2. システム内の異なる Order\_type ごとに、Order\_Type\_Master 構成テーブルに項目を作成し、それにデフォルト・アイコンを割り当てます。

このデータ・アップグレードがステージング・システムで実行されると、どのオーダーも検出されず、従って Order\_Type\_Master テーブルにはデフォルトで提供される "URGENT" しか含まれません。

しかし、同じデータ・アップグレードが実動システムで実行された場合、Order\_Type\_Master テーブルには複数の項目が含まれます。つまり、トランザクション・データベースに含まれるオーダーのタイプごとに 1 つの項目があります。

構成デプロイメント・ツールが再度実行されると、ソースは最新の構成であると想定されるため、すべての新規レコードには削除のマークが付けられます。

その結果、以下のようにして、トランザクション依存の構成データ用にアップグレード・キットまたは Service Pack を設計する必要があります。

1. アップグレード・キット (または Service Pack) には、トランザクション依存の構成データのアップグレード用に入力を準備するためのスクリプトが 1 つ含まれていなければなりません (個別オーダー・タイプの準備リストなど)。その後、このスクリプトを実動データベースで実行できます。また、このスクリプトをテスト・データベースで実行して、2 つの和集合を取ることもできます。
2. アップグレードの次のステップでは、このスクリプトを入力として使用し、それに応じて構成データをアップグレードする必要があります。例えば、ORDER\_TYPE\_MASTER テーブルに挿入します。

構成データに含まれる変更を特定した場合、IBM 技術サポートに連絡してください。



---

## 第 5 章 GUI モードでの CDT のセットアップおよび使用

---

### GUI モードでの CDT のセットアップおよび使用

構成デプロイメント・ツールの設定を構成し、それを GUI から実行できます。

---

### GUI モードの CDT コマンド・ライン引数

構成デプロイメント・ツールを実行する場合、オプションで、表 2 に示されている任意のコマンド・パラメーターを使用できます。

表 2. Java ydk コマンド・ライン引数

引数	説明
IgnoreMissingTables	<p>ソース・データベースとターゲット・データベースを比較する際に、ターゲット・スキーマで欠落している可能性があるテーブルを無視することを指定します。マルチスキーマ・デプロイメントでデータベースを比較する場合、必ず IgnoreMissingTables パラメーターを指定してください。</p> <p>例えば、Test Configuration スキーマを Production Configuration スキーマと比較する場合、YFS_CUSTOMER や YFS_USER などの一部のマスター・テーブルはソース・スキーマにもターゲット・スキーマにも存在しません。この場合、CDT は「テーブルが見つかりません」というエラーをスローします。しかし、-IgnoreMissingTables を Y として渡すことにより、CDT に、欠落しているテーブルを無視させることができます。</p> <p>例:</p> <p>Windows: ydk.cmd -IgnoreMissingTables Y</p> <p>Linux/UNIX: ydk.sh -IgnoreMissingTables Y</p>
ConfigDbDir	<p>config-db.xml および他の関連ファイルを含むディレクトリーを示します。デフォルトでは、ConfigDbDir 引数は、&lt;INSTALL_DIR&gt;/database/cdt ディレクトリーで config-db.xml ファイルを探します。</p> <p>例:</p> <p>Windows: ydk.cmd -ConfigDbDir &lt;directory_path&gt;</p> <p>Linux/UNIX: ydk.sh -ConfigDbDir &lt;directory_path&gt;</p>

---

### GUI モードでの CDT の始動

#### このタスクについて

始める前に、以下を考慮します。

- ほとんどの設定は、特定のソース・データベースとターゲット・データベースの組み合わせに関連しています。ソース・データベースまたはターゲット・データベースのいずれかの名前を変更すると、既存の設定は使用されません。以前の設定に戻すには、以前のソース・データベースおよびターゲット・データベースの名前に戻します。
- 構成デプロイメント・ツールの UI を UNIX サーバー上で Windows クライアントから実行するには、UNIX サーバーに XWindows がインストールされている必要があります。
- Windows コンソールに、WorkBench 始動情報が表示されます。WorkBench の実行中は、コンソールを閉じないでください。コンソールを閉じると、ツールが閉じられ、作業は失われます。

構成デプロイメント・ツールを始動するには、以下のようにします。

## 手順

1. Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench を始動します。Microsoft Windows で、Windows の場合は `ydk.cmd` スクリプト (UNIX/Linux の場合は `ydk.sh`) を `<INSTALL_DIR>%bin` ディレクトリから実行します。
2. 「Development and Deployment WorkBench」メニューで、「ツール (Tools)」 > 「デプロイメント (Deployment)」 > 「構成データ・デプロイメント (Configuration Data Deployment)」を選択します。これにより、構成デプロイメント・ツールの「ログオン (Logon)」ダイアログ・ボックスが開きます。

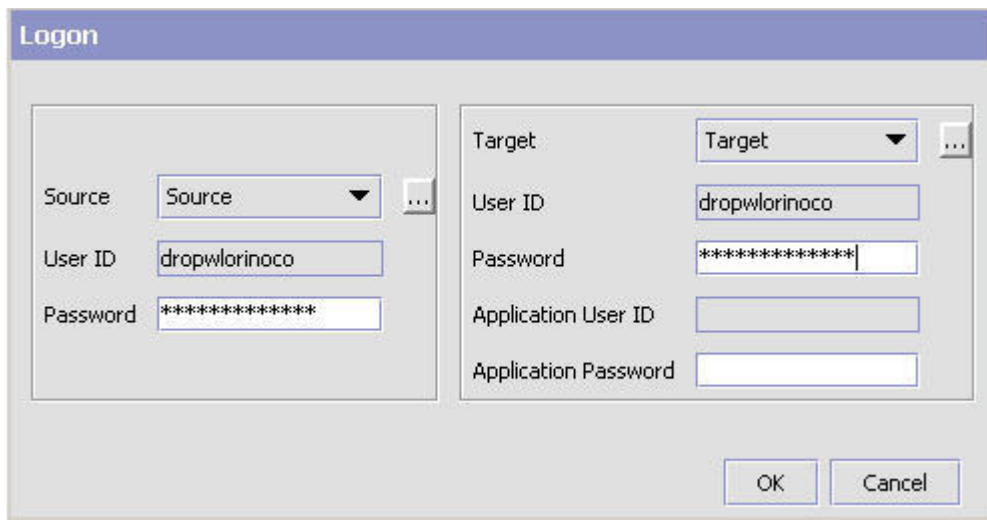


図5. 構成デプロイメント・ツールの「ログオン (Logon)」ダイアログ・ボックス

3. 「ソース」ボタンを選択して、ソース・データベースについて適切な値を入力します。「ターゲット」ボタンを選択して、ターゲット・データベースについて適切な値を入力します。

マルチスキーマ環境では、各構成グループを一度に 1 つずつマイグレーションする必要があります。HUB データおよび組織によって決定される構成データをマイグレーションする場合、これらの 2 つの構成グループはソースである必要

があります。これらのグループが Deployment Explorer に表示され、Sterling Selling and Fulfillment Foundation Sterling Application Platform の一部として提供されます。



図 6. Deployment Explorer ソース - ターゲット・ファイル・システム

構成データは HUB データおよび組織によって決定される構成データ・グループ内にあるため、各グループのターゲットは構成スキーマでなければなりません。組織によって決定されるマスター・データをマイグレーションする場合、ターゲットはマスター・データ・スキーマでなければなりません。また、メタデータをマイグレーションするには、ターゲットがメタデータ・スキーマでなければなりません。

完了したら、ダイアログ・ボックスを閉じます。指定された値は自動的に保存され、あるセッションから次のセッションまで持続します。

「ソース・データベース (Source database)」ウィンドウと「ターゲット・データベース (Target database)」ウィンドウで、表 3 で説明されている、適用可能な値を指定します。

表 3. 構成デプロイメント・ツールの「ログオン (Logon)」ダイアログ・ボックス

フィールド	詳細
Name	論理データベース ID を指定します。ソースの場合、データのコピー元のデータベースを指定します。ターゲットの場合、データの書き込み先のデータベースを指定します。
className	次のようにデータベース・ドライバーのクラス名を指定します。 <ul style="list-style-type: none"> <li>• Oracle を使用している場合、oracle.jdbc.OracleDriver に設定</li> <li>• Microsoft SQL Server 2005/2008 を使用している場合、com.microsoft.sqlserver.jdbc.SQLServerDriver に設定</li> <li>• DB2 を使用している場合、com.ibm.db2.jcc.DB2Driver に設定</li> </ul>

表 3. 構成デプロイメント・ツールの「ログオン (Logon)」ダイアログ・ボックス (続き)

フィールド	詳細
jdbcURL	データベースに接続するための URL を指定します。 <ul style="list-style-type: none"> <li>• Oracle を使用している場合、 jdbc:oracle:thin:@&lt;DatabaseServerHostname/ IPAddress&gt;:&lt;TNSListenerPortNumber&gt;:&lt;DatabaseSID&gt; に設定</li> <li>• Microsoft SQL Server 2005/2008 を使用している場合、 jdbc:sqlserver://&lt;Database Server Hostname&gt;:&lt;Port Number&gt;;DatabaseName=&lt;Database name&gt; に設定</li> <li>• DB2 を使用している場合、jdbc:db2://&lt;Database Server Hostname&gt;:&lt;Port Number&gt;/&lt;Database name&gt; に設定</li> </ul>
dbType	実行しているデータベースのタイプを指定します。次に示すように、すべて小文字で入力してください。 <ul style="list-style-type: none"> <li>• Oracle の場合、oracle と指定</li> <li>• Microsoft SQL Server の場合、sqlserver と指定</li> <li>• DB2 の場合、db2 と指定</li> <li>• XML データ・ソースの場合、xml と指定</li> </ul>
folder	XML データ・ソースを使用している場合、XML ファイルのフォルダの場所の完全パスを指定します。
httpurl	ターゲット・データベースにのみ適用可能です。データがターゲット・データベースにデプロイされた後に、そのデータ・キャッシュがリフレッシュされるアプリケーション・サーバーの URL を指定します。使用する構文は次のとおりです。http://<hostname/ip-address>:<port-number>/<Application>/interop/InteropHttpServlet。ここで、<hostname/IP-address> はアプリケーションが実行されているサーバー・アドレス、<port-number> はアプリケーションが実行されているポート、<Application> はアプリケーションの名前 (例えば、smcfs) です。
httpuser	データがターゲット・データベースにデプロイされた後に、そのデータ・キャッシュがリフレッシュされるアプリケーション・サーバーに関連付けられた姓を指定します。
schema	次のようにスキーマ所有者を指定します。 <ul style="list-style-type: none"> <li>• Oracle または DB2 データベースを使用しており、指定するユーザーが Sterling Selling and Fulfillment Foundation スキーマ所有者と異なる場合は、Sterling Selling and Fulfillment Foundation スキーマの所有者を指定します。</li> <li>• Microsoft SQL Server を使用している場合は、これをブランクのままにします。</li> </ul>
ユーザー	データベースに関連付けられるユーザー名を指定します。

4. 「ログオン (Logon)」ダイアログ・ボックスで、ユーザー名に関連付けられたパスワードを入力します。

「Deployment Explorer」ウィンドウが表示されます。

---

## GUI モードでの CDT の停止

### このタスクについて

構成デプロイメント・ツールを停止するには、以下のようにします。

#### 手順

「Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench」メニューで、「ファイル」 > 「終了」を選択します。これにより、構成デプロイメント・ツールおよび Windows コンソールが閉じます。

---

## GUI モードでの CDT 設定の実行

### このタスクについて

比較操作の動作を決定するプリファレンス (レポート・ディレクトリーなど) およびパラメーターを構成できます。これらのプロパティーを変更する場合、変更は保持されるので、CDT を使用するたびにリセットする必要はありません。これらの変更は、<INSTALL\_DIR>/resources/ydkresources/ydkprefs.xml ファイルに保存されます。

ydkprefs.xml は、初めて CDT を GUI から始動する際に作成されます。あるいは、GUI から CDT を始動しないで、ydkprefs.xml ファイルを手動で作成できます。これにより、ydkprefs.xml ファイルをコマンド・ラインから起動できます。ydkprefs.xml ファイルの手動による作成および編集について詳しくは、29 ページの『テキスト・モードでの CDT 設定の実行』を参照してください。

構成デプロイメント・ツールの設定を指定するには、以下のようにします。

#### 手順


1. 構成デプロイメント・ツールを始動します。構成デプロイメント・ツールの始動について詳しくは、19 ページの『GUI モードでの CDT の始動』を参照してください。
2. 「Deployment Explorer」アクション・バーから、 「プリファレンス」アイコンを選択します。
3. 「プリファレンス」ウィンドウで、表 4 の説明に従って、値を入力します。

表 4. 構成デプロイメント・ツールの設定

コントロール	説明
「設定 (Settings)」タブ	
レポート・ディレクトリー (CDT の場合)	CDT を実行している場合、レポートを生成する絶対パスを指定します。
レポート・ディレクトリー (CDV ツールの場合)	CDV ツールを実行している場合、比較結果レポートをエクスポートする絶対パスを指定します。
カスタム・デプロイメント・クラス	CDT によって処理されないカスタム・テーブルをデプロイするために呼び出す必要があるクラスの名前を指定します。

表 4. 構成デプロイメント・ツールの設定 (続き)

コントロール	説明
表示する変更の最大数 (Max Changes to Display)	表示する差異の最大数を指定します。デフォルトの表示数は 100 です。
監査バージョンのデプロイメント	
古い値の検証 (Validate Old Values)	このチェック・ボックスを選択すると、システムは、ソース・データベースから取得した変更のレコードの属性の予期される古い値と、ターゲット・データベースでの対応するレコードの現行値の比較に基づいて競合を検出します。
ロック ID の検証 (Validate Lockid)	このチェック・ボックスを選択すると、システムは、ソース・データベースから取得した変更のレコードの予期されるロック ID 値と、ターゲット・データベースでの対応するレコードの現行ロック ID の比較に基づいて競合を検出します。
削除の前にレコードの存在を検証 (Validate Record Exists Before Delete)	このチェック・ボックスを選択すると、システムは、レコードの削除を試行する前に、ターゲット・データベース内でそのレコードの存在を検証します。レコードが存在しない場合、操作には競合というマークが付けられます。
「変換」タブ	
テーブル・エレメント	追加または削除できるテーブル。
テーブル名属性 (Table Name Attribute)	変換を実行するテーブルの名前を指定します。構文および大/小文字の区別は、Sterling Selling and Fulfillment Foundation ERD で使用されるテーブルの名前と一致している必要があります。カスタム・テーブルは変換できません。「詳細」アイコンを選択して、値を指定してください。
列エレメント (Column Element)	追加または削除できる列。
列名属性 (Column Name Attribute)	変換するデータが含まれる列の名前を指定します。構文および大/小文字の区別は、Sterling Selling and Fulfillment Foundation ERD で使用される列の名前と一致している必要があります。拡張された列は変換できます。「詳細」アイコンを選択して、値を指定してください。
変換エレメント (Transform Element)	この列に対する変換を定義します。列ごとに、1 つ以上の変換を定義できます。これらの変換は、実行された順番にこの列のデータに適用されます。削除アクションを使用して、親エレメントを削除することにより、列ごとに複数の変換を指定できます。
マッチング属性 (Match Attribute)	ソース・データで検索するパターンを指定します。このパターンに一致するすべての検索結果は、「置換属性 (Replace attribute)」で指定された値に置き換えられます。「詳細」アイコンを選択して、値を指定してください。
置換属性 (Replace Attribute)	パターンを置き換える値を指定します。「詳細」アイコンを選択して、値を指定してください。

表 4. 構成デプロイメント・ツールの設定 (続き)

コントロール	説明
XPath 属性 (XPath Attribute)	<p>条件付き。変換する列に XML 以外のデータが含まれる場合、この XPath 属性を指定する必要はありません。しかし、Sterling Selling and Fulfillment Foundation の一部の構成情報は、XML としてデータベースに保管されます。</p> <p>変換する列に XML データが含まれる場合、この属性を使用して、変換する正確な属性の場所を指定します。</p> <p>構文として <code>xml:/Configuration/Connection/Host/@IPAddress</code> を使用してください。「詳細」アイコンを選択して、値を指定してください。</p>
<b>追加専用テーブル (Append-Only Tables) タブ</b>	
追加専用テーブル (Append-only Tables)	<p>一部の行が Sterling Selling and Fulfillment Foundation の外部のデータを保持している構成テーブルがあれば、指定します。これにより、デプロイメント中にデータが削除されなくなります。そのテーブルとすべての従属テーブルを指定してください。</p> <p><b>注:</b> 外部で管理される行は、ソース・データベースに含まれていてはなりません。含まれていると、予期しない結果になる可能性があります。</p>
<b>テーブルの無視 (Ignore Tables) タブ</b>	
テーブルの無視 (Ignore Tables)	<p>ツールによりソースからターゲットにデプロイしたくない外部構成テーブルを指定します。テーブルを無視すると、すべての従属テーブルも自動的に無視されます。</p>





---

## 第 6 章 テキスト・モードでの CDT のセットアップおよび使用

---

### テキスト・モードでの CDT のセットアップおよび使用

構成デプロイメント・ツールの設定は、コマンド・ラインから実行および構成できます。

---

### テキスト・モードの CDT コマンド・ライン引数

構成デプロイメント・ツールを実行する場合、オプションで、表 5 に示されている任意のコマンド・パラメーターを使用できます。

表 5. Java *cdtshell* スクリプト・コマンド・ライン引数

引数	説明
ColonyId	比較またはデプロイするコロニー ID を指定します。コロニーは、一度に 1 つのみ渡すことができます。  例:  Windows: <code>cdtshell.cmd -ColonyId E1</code>  Linux/UNIX: <code>cdtshell.sh -ColonyId E1</code>
IgnoreMissingTables	ソース・データベースとターゲット・データベースを比較する際に、ターゲット・スキーマで欠落している可能性があるテーブルを無視することを指定します。マルチスキーマ・デプロイメントでデータベースを比較する場合、必ず <code>IgnoreMissingTables</code> 引数を指定してください。  例えば、 <code>Test Configuration</code> スキーマを <code>Production Configuration</code> スキーマと比較する場合、 <code>YFS_CUSTOMER</code> や <code>YFS_USER</code> などの一部のマスター・テーブルはソース・スキーマにもターゲット・スキーマにも存在しません。この場合、CDT は「テーブルが見つかりません」というエラーをスローします。しかし、 <code>-IgnoreMissingTables</code> を <code>Y</code> として渡すことにより、CDT に、欠落しているテーブルを無視させることができます。  例:  Windows: <code>cdtshell.cmd -IgnoreMissingTables Y</code>  Linux/UNIX: <code>cdtshell.sh -IgnoreMissingTables Y</code>

表 5. Java cdtshell スクリプト・コマンド・ライン引数 (続き)

引数	説明
LabelId	<p>ラベルの作成に使用するラベル ID の値を指定します。例えば、デプロイメントの前は BEGIN_&lt;LabelId&gt;、デプロイメントの後は END_&lt;LabelId&gt; となります。この引数が渡されない場合、ラベルは作成されません。</p> <p>例:</p> <p>Windows: cdtshell.cmd -LabelId OrgA1</p> <p>Linux/UNIX: cdtshell.sh -LabelId OrgA1</p>
CompareOrganizationCode	<p>config-db.xml で定義されているとおり、渡された組織コードに基づいて、比較する組織を指定します。CompareOrganizationCode 引数を渡さない場合、対応するフィルターは使用されません。</p> <p>コンマで区切られたフォーマットを使用して、比較する組織を指定します (例:Org-1,Org-2)。 CompareOrganizationCode 引数を使用し、組織を指定しない場合、CDT はすべての組織を比較します。参加するすべての組織をまとめて比較することが推奨されています。例えば、2 つの組織、Org-01 および Org-02 が、2 つの出荷ノード、Node-01 および Node-02 に参加する場合、Org-1,Org-2,Node-01,Node-02 と指定します。</p> <p>例:</p> <p>Windows: ydk.cmd -CompareOrganizationCode Org-1,Org-2,Node-01,Node-02</p> <p>Linux/UNIX: ydk.sh -CompareOrganizationCode Org-01,Org-02,Node-01,Node-02</p>

## テキスト・モードでの CDT の実行

### このタスクについて

このセクションでは、テキスト・ベース (GUI 以外) のインターフェースを使用して、構成デプロイメント・ツールをコマンド・ラインから実行する方法を説明します。

構成デプロイメント・ツールをコマンド・ラインから実行するには、以下のようにします。

### 手順

- Windows の場合、<INSTALL\_DIR>/bin/cdtshell.cmd.in スクリプト (UNIX/Linux の場合、cdtshell.sh.in) を編集し、以下のプロパティーを設定します。

プロパティー  
説明

### **SOURCE\_DB**

ソース・データベースの論理 ID。この値は、ydkprefs.xml ファイルの <SourceDatabases> エレメントの Name 設定に対応します。

### **TARGET\_DB**

ターゲット・データベースの論理 ID。この値は、ydkprefs.xml ファイルの <TargetDatabases> エレメントの Name 設定に対応します。

### **SOURCE\_PASSWORD**

ソース・データベースに必須のパスワードを指定します。

**注:** ソースが XML フォルダの場合、このパラメーターは設定する必要がありません。

### **TARGET\_PASSWORD**

ターゲット・データベースに必須のパスワードを指定します。

**注:** ソースが XML フォルダの場合、このパラメーターは設定する必要がありません。

### **TARGET\_HTTP\_PASSWORD**

サーバー上のキャッシュをリフレッシュするために必要な http ユーザーのパスワードを指定します。

**注:** このパラメーターは、サーバー上にリフレッシュするキャッシュがない場合には不要です。

- Windows の場合、<INSTALL\_DIR>/bin/setupfiles.cmd スクリプトを実行します (UNIX/Linux の場合は、setupfiles.sh)。
- 比較操作のために、CDT プリファレンスおよびパラメーターを構成します。ydkprefs.xml ファイルは、CDT プリファレンスを判別し、GUI から CDT を実行する際に自動的に作成されます。ただし、CDT GUI を使用せずに、ydkprefs.xml ファイルを手動で作成し、編集することも可能です。ydkprefs.xml ファイルの手動による作成および編集については、『テキスト・モードでの CDT 設定の実行』を参照してください。
- Windows の場合、cdtshell.cmd スクリプト (UNIX/Linux の場合、cdtshell.sh) を <INSTALL\_DIR>%bin ディレクトリーから実行します。

---

## **テキスト・モードでの CDT 設定の実行**

### **このタスクについて**

ydkprefs.xml.sample ファイルを編集し、そのファイルを ydkprefs.xml として保存することにより、比較操作の動作を決定するプリファレンスおよびパラメーター設定を手動で構成できます。

構成デプロイメント・ツールの設定を構成するには、以下のようにします。

### **手順**

- <INSTALL\_DIR>/resources/ydkresources/ydkprefs.xml.sample ファイルを ydkprefs.xml に名前変更します。

2. ydkprefs.xml で、CDT プリファレンス・エレメントのパラメーターを指定します。表 6 に、ydkprefs.xml のエレメントが説明されています。

表 6. 構成デプロイメント・ツールの設定

エレメント	説明
<b>&lt;Settings&gt; エレメント</b>	
CustomEntityClass	CDT によって処理されないカスタム・テーブルをデプロイするために呼び出す必要があるクラスの名前を指定します。 <b>注:</b> このフィールドを使用しない場合、このフィールドの値をヌルに設定します。例えば、CustomEntityClass="" のようにします。
MaxChangesToDisplay	表示する差異の最大数を指定します。デフォルトの表示数は 100 です。 <b>注:</b> このフィールドを使用しない場合、このフィールドの値をヌルに設定します。例えば、MaxChangesToDisplay="" のようにします。
ReportsDir	レポートを生成する絶対パスを指定します。 <b>注:</b> このフィールドを使用しない場合、このフィールドの値をヌルに設定します。例えば、ReportsDir="" のようにします。
ValidateLockid	システムがソース・データベースから取得した変更のレコードの予期されるロック ID と、ターゲット・データベースでの対応するレコードの現行ロック ID の比較に基づいて競合を検出するようにしたい場合は、Y を指定します。  デフォルトでは、N に設定されます。
ValidateOldValues	システムがソース・データベースから取得した変更のレコードの属性の予期される古い値と、ターゲット・データベースでの対応するレコードの現行値の比較に基づいて競合を検出するようにしたい場合は、Y を指定します。  デフォルトでは、Y に設定されます。
ValidateRecordExistsBeforeDelete	システムがレコードの削除を試行する前に、ターゲット・データベース内でそのレコードの存在を検証するようにしたい場合は、Y を指定します。レコードが存在しない場合、操作には競合というマークが付けられます。  デフォルトでは、Y に設定されます。
<b>&lt;SourceDatabases&gt; エレメント</b>	
Name	論理データベース ID を指定します。ソースの場合、データのコピー元のデータベースを指定します。ターゲットの場合、データの書き込み先のデータベースを指定します。
className	次のようにデータベース・ドライバーのクラス名を指定します。 <ul style="list-style-type: none"> <li>• Oracle を使用している場合、oracle.jdbc.OracleDriver に設定</li> <li>• Microsoft SQL Server 2005/2008 を使用している場合、com.microsoft.sqlserver.jdbc.SQLServerDriver に設定</li> <li>• DB2 を使用している場合、com.ibm.db2.jcc.DB2Driver に設定</li> </ul>
dbType	実行しているデータベースのタイプを指定します。次に示すように、すべて小文字で入力してください。 <ul style="list-style-type: none"> <li>• Oracle の場合、oracle と指定</li> <li>• Microsoft SQL Server の場合、sqlserver と指定</li> <li>• DB2 の場合、db2 と指定</li> <li>• XML データ・ソースの場合、xml と指定</li> </ul>

表 6. 構成デプロイメント・ツールの設定 (続き)

エレメント	説明
folder	XML データ・ソースを使用している場合、XML ファイルのフォルダーの場所の完全パスを指定します。 注: ソース・データベースがデータベース・スキーマである場合、この属性の値をヌルに設定する必要があります。例えば、folder="" のようにします。
jdbcURL	データベースに接続するための URL を指定します。 <ul style="list-style-type: none"> <li>• Oracle を使用している場合、jdbc:oracle:thin:@&lt;DatabaseServerHostname/IPAddress&gt;:&lt;TNSListenerPortNumber&gt;:&lt;DatabaseSID&gt; に設定</li> <li>• Microsoft SQL Server 2005/2008 を使用している場合、jdbc:sqlserver://&lt;Database Server Hostname&gt;:&lt;Port Number&gt;;DatabaseName=&lt;Database name&gt; に設定</li> <li>• DB2 を使用している場合、jdbc:db2://&lt;Database Server Hostname&gt;:&lt;Port Number&gt;/&lt;Database name&gt; に設定</li> </ul>
schema	次のようにスキーマ所有者を指定します。 <ul style="list-style-type: none"> <li>• Oracle または DB2 データベースを使用しており、指定するユーザーが Sterling Selling and Fulfillment Foundation スキーマ所有者と異なる場合は、Sterling Selling and Fulfillment Foundation スキーマの所有者を指定します。</li> <li>• Microsoft SQL Server を使用している場合は、このフィールドをブランクのままにします。例えば、schema="" のようにします。</li> </ul> 注: Oracle および DB2 データベースの場合、スキーマ名がユーザー名と同じ場合は、このフィールドをブランクのままにします。
ユーザー	データベースに関連付けられるユーザー名を指定します。
<TargetDatabases> エレメント	
Name、className、dbType、folder、jdbcURL、schema、および user 属性の説明については、この表の <SourceDatabases> エレメントのセクションを参照してください。	
httpurl	ターゲット・データベースにのみ適用可能です。データがターゲット・データベースにデプロイされた後に、そのデータ・キャッシュがリフレッシュされるアプリケーション・サーバーの URL を指定します。使用する構文は次のとおりです。http://<hostname/ip-address>:<port-number>/<Application>/interop/InteropHttpServlet。ここで、<hostname/IP-address> はアプリケーションが実行されているサーバー・アドレス、<port-number> はアプリケーションが実行されているポート、<Application> はアプリケーションの名前 (例えば、smcfs) です。
httpuser	データがターゲット・データベースにデプロイされた後に、そのデータ・キャッシュがリフレッシュされるアプリケーション・サーバーに関連付けられたユーザー名を指定します。
<SourceTargetPrefs> エレメント	
SourceDatabase	ソース・データベースの名前を指定します。
TargetDatabase	ターゲット・データベースの名前を指定します。
<Transformations> エレメント	
追加または削除できるテーブルおよび列。	

表 6. 構成デプロイメント・ツールの設定 (続き)

エレメント	説明
<Table> エレメントの Name 属性	<p>変換を実行するテーブルの名前を指定します。構文および大/小文字の区別は、Sterling Selling and Fulfillment Foundation ERD で使用されるテーブルの名前と一致している必要があります。カスタム・テーブルは変換できません。「詳細」アイコンを選択して、値を指定してください。</p> <p><b>注:</b> このフィールドを使用しない場合、このフィールドの値をヌルに設定します。例えば、Name="" のようにします。</p>
<Column> エレメントの Name 属性	<p>変換するデータが含まれる列の名前を指定します。構文および大/小文字の区別は、Sterling Selling and Fulfillment Foundation ERD で使用される列の名前と一致している必要があります。拡張された列は変換できます。「詳細」アイコンを選択して、値を指定してください。</p> <p><b>注:</b> このフィールドを使用しない場合、このフィールドの値をヌルに設定します。例えば、Name="" のようにします。</p>
<Transform> エレメント	
<p>この列に対する変換を定義します。列ごとに、1 つ以上の変換を定義できます。これらの変換は、実行された順番にこの列のデータに適用されます。削除アクションを使用して、親エレメントを削除することにより、列ごとに複数の変換を指定できます。</p>	
マッチング	<p>ソース・データで検索するパターンを指定します。このパターンに一致するすべての検索結果は、「置換属性 (Replace attribute)」で指定された値に置き換えられます。「詳細」アイコンを選択して、値を指定してください。</p>
置換	<p>パターンを置き換える値を指定します。「詳細」アイコンを選択して、値を指定してください。</p>
XPath	<p>条件付き。変換する列に XML 以外のデータが含まれる場合、この XPath 属性を指定する必要はありません。しかし、Sterling Selling and Fulfillment Foundation の一部の構成情報は、XML としてデータベースに保管されます。</p> <p>変換する列に XML データが含まれる場合、この属性を使用して、変換する正確な属性の場所を指定します。</p> <p>構文として <code>xml:/Configuration/Connection/Host/@IPAddress</code> を使用してください。「詳細」アイコンを選択して、値を指定してください。</p>
<Ignore> エレメント	
<Table> エレメントの Name 属性	<p>ツールによりソースからターゲットにデプロイしたくない外部構成テーブルを指定します。テーブルを無視すると、すべての従属テーブルも自動的に無視されます。</p> <p><b>注:</b> このフィールドを使用しない場合、このフィールドの値をヌルに設定します。例えば、Name="" のようにします。</p>
<AppendOnly> エレメント	
<Table> エレメントの Name 属性	<p>一部の行が Sterling Selling and Fulfillment Foundation の外部のデータを保持している構成テーブルがあれば、指定します。これにより、デプロイメント中にデータが削除されなくなります。そのテーブルとすべての従属テーブルを指定してください。</p> <p><b>注:</b> 外部で管理される行は、ソース・データベースに含まれてはなりません。含まれていると、予期しない結果になる可能性があります。</p> <p><b>注:</b> このフィールドを使用しない場合、このフィールドの値をヌルに設定します。例えば、Name="" のようにします。</p>

## 第 7 章 CDT を使用したエレメントの変換について

### CDT を使用したエレメントの変換について

データのあるデータベース・インスタンスから別のデータベース・インスタンスにデプロイする場合、特定のデータ・エレメントの値をオーバーライドできます。例えば、ソース環境とターゲット環境のネットワーク設定 (ホスト名、ポート番号、および IP アドレス) が異なる場合、構成デプロイメント・ツールは、ターゲット環境に適したものになるように設定を変換できます。

変換はパターン・マッチングとして実行され、ソース・データベースのデータを置換してから、ターゲット・データベースにデプロイします。

マッチングと置換はストリング・リテラル全体に対して実行され、文字のワイルドカード検索は許可されません。

例えば、ソース・データベースの以下のような構成 XML について考えてみましょう。

```
<SubFlowConfig>
  <Link>
    <Properties DeliveryMode=""
      InitialContextFactory="weblogic.jndi.WLInitialContextFactory"
      ProviderURL="t3://localhost:7001" QCFLookup="TEST_AGENT_QCF"
      QName="DefaultAgentQueue" TimeToLive=""/>
  </Link>
</SubFlowConfig>
```

ターゲット・データベースには、ポート番号の値として 7221 があり、QCFLookup として AGENT\_QCF があります。これらがソース・データベースの値でオーバーライドされないようにします。これらの値を変換するには、表 7 の説明に従って、値を指定します。

表 7. エレメントの変換

エレメント	属性	値
ProviderURL		
	マッチング	7001 (または、0 の出現箇所をすべて検索するには 0) を指定します。
	置換	7221 (または 2 の出現箇所をすべて置換するには 2) を指定します。
	XPath	xml:/SubFlowConfig/Link/Properties/@ProviderURL
QCFLookup		
	マッチング	TEST_ の出現箇所をすべて検索するには、TEST_ を指定します。
	置換	TEST_ が削除されるようにするには、ブランクのままにします。
	XPath	xml:/SubFlowConfig/Link/Properties/@QCFLookup

この例を使用すると、「変換」タブは、図7 に示すようになります。

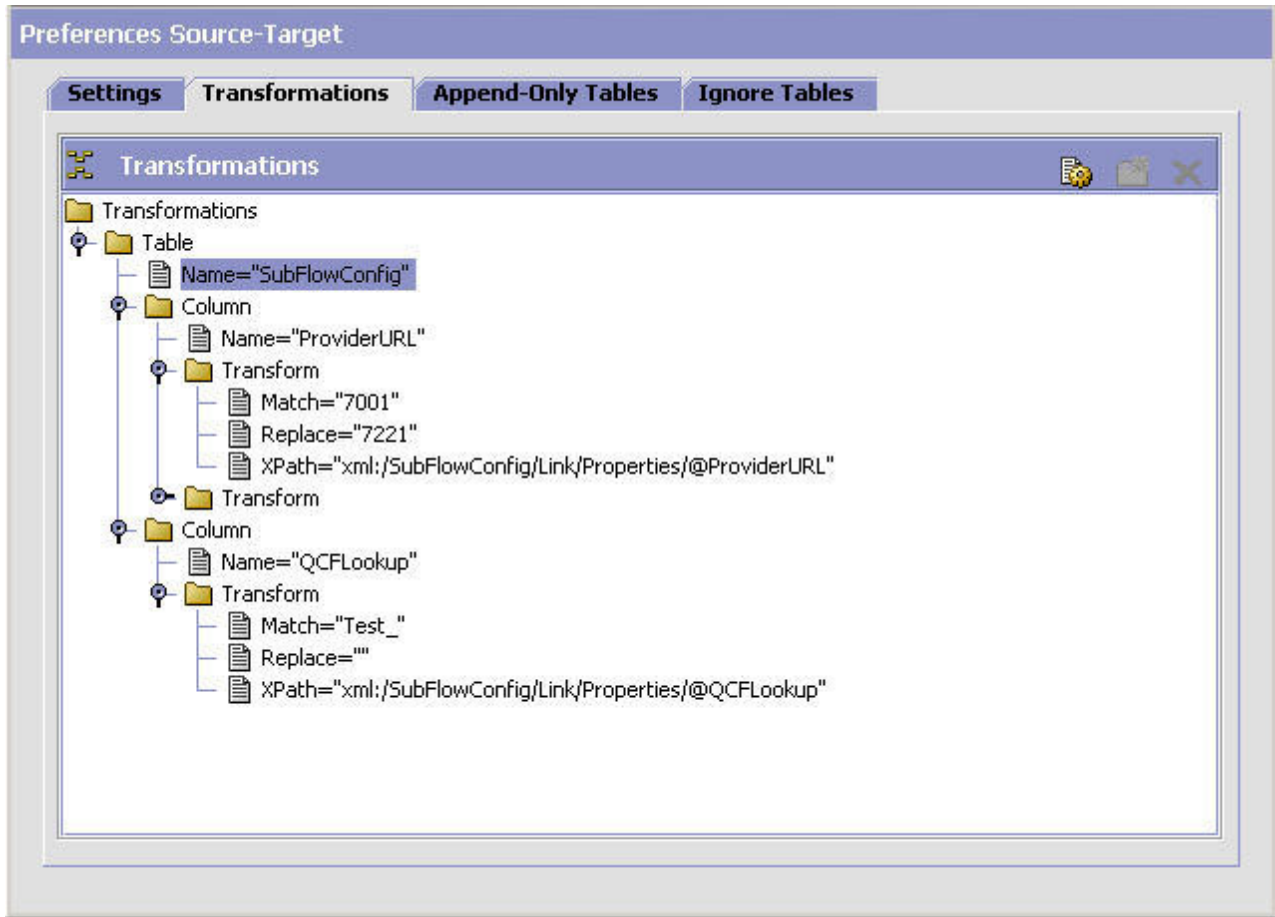



図7. 変換の例

## CDT を使用したエレメントの変換

### このタスクについて

構成データのエレメントを変換するには、以下のようにします。

#### 手順

1. 「Deployment Explorer」ウィンドウのアクション・バーから、 「プリファレンス」アイコンを選択します。
2. 「プリファレンス」ウィンドウで、「変換」タブを選択し、値を入力します。データをデプロイするとき、指定したそれらの変換値が構成データとともにデプロイされます。

データをデプロイする前に、まず 35 ページの『CDT のソース・データベースとターゲット・データベースの比較』の説明に従って、データベースの比較を実行する必要があります。



---

## 第 8 章 CDT のソース・データベースとターゲット・データベースの比較

---

### CDT のソース・データベースとターゲット・データベースの比較

実稼働環境に構成データをデプロイするには、2 つのデータベースを比較し、その後で変更をデプロイします。

注: CDT は、ソース環境とターゲット環境の両方がデータベースである場合、特殊文字をデータと見なします。

比較が完了した後、以下のいずれかの作業を実行できます。

- 38 ページの『CDT を使用したデータベースの差異の検証』の説明に従って、差異を検証します。
- 40 ページの『CDT を使用した比較結果のエクスポート』の説明に従って、レポートをエクスポートします。
- 41 ページの『CDT を使用した差異のレポートの生成』の説明に従って、レポートを生成します。
- 45 ページの『CDT を使用した構成データのデプロイ』の説明に従って、変更をデプロイします。

---

### CDT を使用したソース・データとターゲット・データの比較

#### このタスクについて




初めて データをデプロイする場合、データベース全体を比較してください。ソース・データベースとターゲット・データベースの間の差異が比較的少ないことを確認した後にのみ、少しずつ比較してください。


ソース・データベースとターゲット・データベースを比較するには、以下のようになります。

#### 手順

1. Deployment Explorer ツリーで、比較するデータを選択します。

使用可能な比較グループは以下のとおりです。

- データベース全体の場合、 Sterling Selling and Fulfillment Foundation 「構成」アイコンを選択します。
- 特定の構成グループの場合、 「構成グループ (Configuration Group)」アイコンを選択します。
- 特定のドライバー・エンティティの場合、 「ドライバー・エンティティ (Driver Entity)」アイコンを選択します。

2. 「Deployment Explorer」アクション・バーから、 「比較 (Compare)」アイコンを選択します。「比較結果 (Comparison Results)」ウィンドウが、すべての差異をリストして、右上に表示されます。「比較結果ステータス (Comparison Results Status)」パネルが右下に表示されます。

CompareOrganizationCode 引数を ydk コマンドの一部として指定しない場合、37 ページの『CDT を使用した比較対象組織の指定』で説明されているように、CDT は比較する組織を指定するためのプロンプトを出します。

さらに、マルチスキーマ・デプロイメントでデータを比較する場合、38 ページの『CDT を使用したマルチスキーマ環境でのデータの比較について』で説明されているように、CDT はコロニー ID を指定するためのプロンプトを出します。

## タスクの結果

**注:** 1 つのテーブルが複数のグループに含まれている場合、または複数のエンティティに属している場合、その差異は、合計数では複数回カウントされる可能性があります。

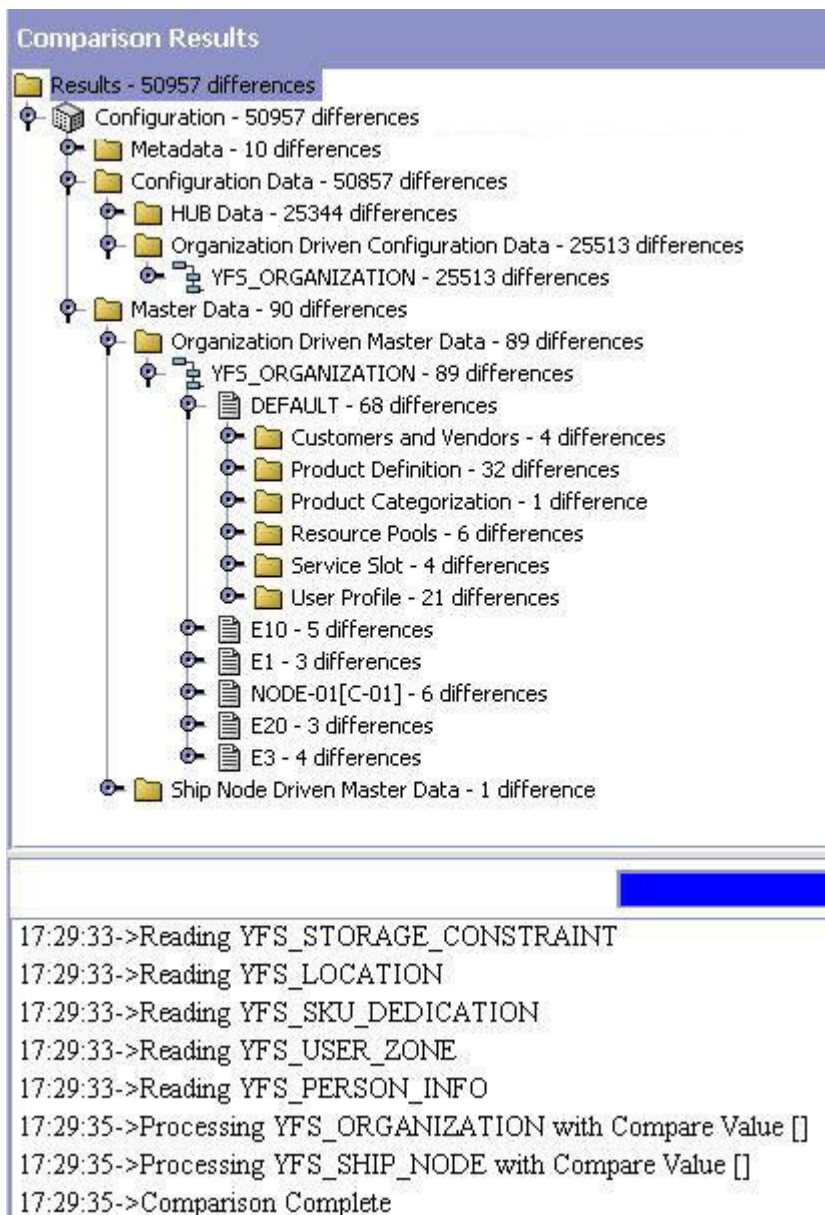

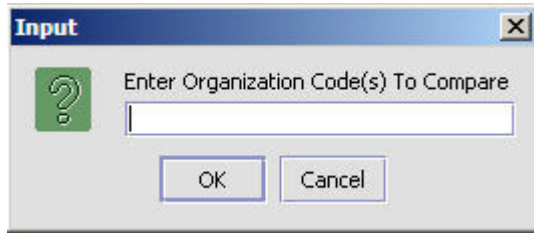


図 8. 「比較結果 (Comparison Results)」 ウィンドウ

## CDT を使用した比較対象組織の指定

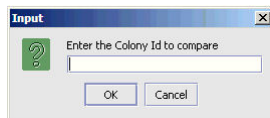
CompareOrganizationCode 引数を ydk コマンドの一部として渡さない場合、CDT は「Deployment Explorer」ウィンドウで  「比較 (Compare)」アイコンを選択すると、「入力」ポップアップ・ウィンドウを表示します。「入力」ポップアップ・ウィンドウに、比較する組織を指定するためのプロンプトが出されます。



コンマで区切られたフォーマットを使用して、比較する組織を指定します (例:Org-1,Org-2)。組織を入力しないで、「OK」をクリックすると、CDT はすべての組織を比較します。参加するすべての組織をまとめて比較することが推奨されています。例えば、2 つの組織、Org-01 および Org-02 が、2 つの出荷ノード、Node-01 および Node-02 に参加する場合、Org-1,Org-2,Node-01,Node-02 と指定します。

## CDT を使用したマルチスキーマ環境でのデータの比較について

マルチスキーマ環境でデータを比較する場合、CDT は比較するコロニー ID を入力するためのプロンプトを表示します。



コロニー ID を入力しないで「OK」をクリックすると、CDT はすべてのコロニーを比較します。

コロニー ID を入力して、「OK」をクリックすると、CDT は当該コロニー (つまりコロニーに属する一連の組織) のデータのみ比較します。構成スキーマに比較したい他のコロニーが含まれる場合、各コロニーを個別に比較する必要があります。例えば、Colony\_01 と Colony\_02 を別々に比較する必要があります。

マスター・データ・スキーマでデータを比較する場合、CDT はコロニー ID を入力するためのプロンプトを表示する代わりに、「入力」ポップアップ・ウィンドウを表示します。なぜなら、マスター・データはコロニー全体のデータだからです。

組織の名前を「入力」ポップアップ・ウィンドウに入力すると、CDT は特定のコロニーに属する組織のデータのみ比較します。

組織を「入力」ポップアップ・ウィンドウに入力しないで「OK」をクリックすると、CDT はすべてのコロニーを比較します。


---

## CDT を使用したデータベースの差異の検証


### このタスクについて


データベース間の差異を検証するには、以下のようにします。


### 手順


1.  を選択して、2 つのデータベースを比較します。データベースの比較について詳しくは、35 ページの『CDT を使用したソース・データとターゲット・データの比較』を参照してください。


2. 「比較結果 (Comparison Results)」ツリーで、対応するエンティティを展開し、検証するテーブルを選択します。

3. 「比較結果 (Comparison Results)」アクション・バーから、 を選択します。比較の結果の範囲は以下のとおりです。

 - 「未変更」アイコンは、エンティティに、差異がある従属テーブルが含まれることを示します。

 - 「追加 (Add)」アイコンは、レコードがターゲット・データベースに挿入されることを示します。

 - 「削除」アイコンは、レコードがターゲット・データベースから削除されることを示します。

 - 「変更」アイコンは、ターゲット・データベースで列が更新されることを示します。

4. ステップ 3 の説明に従って、エンティティのアイコンを選択して、そのエンティティの「レコードの詳細 (Record Details)」ウィンドウを表示します。ウィンドウには、ターゲット・データベースのエンティティの値に関するデータが表示されます。例えば、変更されたデータの「レコードの詳細 (Record Details)」ウィンドウには、以下のセクションが表示されます。

- 上部セクション - ターゲット・データベースで変更された値が表示されます。
- 下部セクション - 未変更の値が表示されます。

40 ページの図 9 は、挿入されたデータの「レコードの詳細 (Record Details)」ウィンドウを示しています。

Record Details	
Name	Inserted
Name	Value
BaseProcessTypeListenedTo	
TimeTriggerable	Y
BaseTranid	CHANGE_COUNT_REQ_STATUS
ExternallyTriggerable	Y
UserTriggerable	Y
WorksOffTaskQ	Y
DropStatusFilter	EXTN
CanResolveHoldType	N
OwnerKey	DEFAULT
ListenerType	
RequiresChainedDocType	N
SingleDropStatus	N
BaseTranname	Change Count Request Status
AgentJavaClass	
BaseTransactionKey	CHANGE_COUNT_REQ_STATUS
SupportedDependencyType	N
SupportedCompletionType	N
BaseProcessTypeKey	COUNT_EXECUTION
HoldTypeEnabled	N

図9. 「レコードの詳細 (Record Details)」 ウィンドウ

5. データを検証した後、41 ページの『CDT を使用した差異のレポートの生成』の説明に従って、これらの差異のレポートを生成することができます。

## CDT を使用した比較結果のエクスポート


### このタスクについて

構成の差異を、後で比較するために、または既存の構成のバックアップとしてエクスポートできます。

比較結果を XML ファイルにエクスポートするには、以下のようになります。

### 手順

1. Sterling Selling and Fulfillment Foundation 構成デプロイメント・ツールの「プリファレンス」の「設定 (Settings)」タブにある「レポート・ディレクトリー (Reports Directory)」フィールドに、比較レポートが生成されるディレクトリーの位置が指定されていることを確認します。これらの設定の指定については、23 ページの『GUI モードでの CDT 設定の実行』を参照してください。

2. 「比較結果 (Comparison Results)」アクション・バーから、 を選択します。Windows エクスプローラーで、「レポート・ディレクトリー (Reports Directory)」フィールドに指定されている位置を参照します。

Sterling Selling and Fulfillment Foundation 構成デプロイメント・ツールは、サブディレクトリーをこのディレクトリーに自動的に作成します。例えば、D:/reports を「レポート・ディレクトリー (Reports Directory)」フィールドに指定し、比較結果を 2008 年 5 月 23 日午後 3 時 40 分にエクスポートした場合、CDT は D:/reports/export20080523154024 というサブディレクトリーを作成します。この新規のサブディレクトリーには、ydkexport.xml ファイルが含まれ、そこに比較結果が入っています。

インポートが完了した後、以下のいずれかの作業を実行できます。

- 38 ページの『CDT を使用したデータベースの差異の検証』の説明に従って、差異を検証します。
- 『CDT を使用した差異のレポートの生成』の説明に従って、差異のレポートを作成します。
- 45 ページの『構成データのデプロイ』の説明に従って、変更をデプロイします。

---




## CDT を使用した差異のレポートの生成

### このタスクについて

ソース・データベースとターゲット・データベースの間の差異のレポートを生成できます。

差異のレポートを生成するには、以下のようにします。

### 手順

1.  を選択して、reports ディレクトリーを指定したことを確認します。 CDT の設定の指定について詳しくは、23 ページの『GUI モードでの CDT 設定の実行』を参照してください。
2.  を選択して、2 つのデータベースを比較します。データベースの比較について詳しくは、35 ページの『CDT を使用したソース・データとターゲット・データの比較』を参照してください。
3. 「比較結果 (Comparison Results)」ツリーで、「結果」ノードを選択します。
4. 「比較結果 (Comparison Results)」アクション・バーから、 を選択します。「ステータス」パネルに、レポート生成プロセスが成功したかどうかを判断できるトレース・メッセージが表示されます。レポートの作成が正常に行われたというメッセージとともに、レポートの場所が表示されます。
5. Windows エクスプローラーで、レポート・ディレクトリーを参照します。構成デプロイメント・ツールは、作成時間に基づいて命名したサブディレクトリーを、指定したディレクトリーに作成します。

例えば、レポート・ディレクトリーとして D:/reports を指定し、2008 年 5 月 23 日午後 3 時 40 分にレポートを生成した場合、CDT は D:/reports ディレクトリーに 20080523154024 というサブディレクトリーを作成します。

この新規のサブディレクトリーには、以下が含まれます。

- UI で表示される、変更の全体のサマリーが含まれる index.xml ファイル
- テーブルごとに 1 つの XML ファイル (これには、変更と各変更の詳細が含まれます)

6. XML ファイルを開いて、差異を表示します。

## タスクの結果

別のレポートを生成する場合、新規のディレクトリーが作成され、別の XML ファイルのセットがそこに生成されます。

---

## CDT を使用した構成の差異のインポート

### このタスクについて

比較結果をエクスポートすることによって得られた構成の差異をインポートできます。

構成デプロイメント・ツールは、データベースの比較、比較結果のエクスポート、または比較結果のインポート時に、特殊文字を含むデータをサポートしません。

比較結果のエクスポートについて詳しくは、40 ページの『CDT を使用した比較結果のエクスポート』を参照してください。

構成の差異をインポートするには、以下のようになります。

### 手順

1. <INSTALL\_DIR>/bin/ydk.cmd スクリプトを実行します。このスクリプトにより、Microsoft Windows コンソールが開き、Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench が始動します。
2. 「Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench」メニューで、「ツール (Tools)」 > 「デプロイメント (Deployment)」 > 「結果のインポート (Import Results)」を選択します。Sterling Selling and Fulfillment Foundation 構成デプロイメント・ツールの「インポート」ダイアログ・ボックスが表示されます。



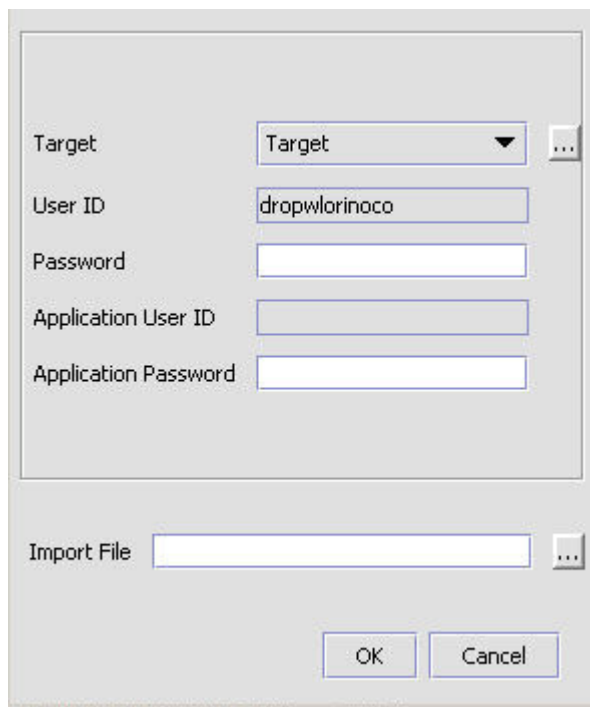


図 10. 「インポート」ダイアログ・ボックス

3. 「ターゲット」ボタンを選択して、ターゲット・データベースについて適切な値を入力します。「ファイルのインポート」を選択して、インポートするファイルのパスを入力します。

完了したら、「OK」をクリックして、ダイアログ・ボックスを閉じます。

4. 比較結果がロードされた後、以下のいずれかの作業を実行できます。
  - 38 ページの『CDT を使用したデータベースの差異の検証』で示されている説明に従って、差異を検証します。
  - 41 ページの『CDT を使用した差異のレポートの生成』で示されている説明に従って、レポートを生成します。
  - 42 ページの『CDT を使用した構成の差異のインポート』で示されている説明に従って、変更をデプロイします。



---

## 第 9 章 構成データのデプロイ

---

### 構成データのデプロイ

構成データをデプロイする前に、データを比較し、差異を調べることにより、必ず正しいデータがデプロイされるようにしてください。

さらに、ロールバックの問題に対応済みであることを確認してください。 Sterling Selling and Fulfillment Foundation は、サンプルのバックアップおよびロールバック・スクリプトを提供します。これらのスクリプトについては、66 ページの『データ・ロールバック・スクリプト』を参照してください。

**ヒント:** 初めてデータをデプロイする場合、データベース全体をデプロイしてください。ソース・データベースとターゲット・データベースの間の差異が比較的少ないことが確かな場合のみ、少しずつデプロイしてください。

**注:** 出荷ノードを比較する場合、ソース・スキーマ内のノードに調整理由コード値 PACK、SHIP、または RECEIVE がない場合でも、これらの値はターゲット・スキーマ内のノードのデフォルト値になります。


---

### CDT を使用した構成データのデプロイ





#### このタスクについて

構成データをデプロイするには、以下のようになります。

#### 手順

1. 2 つのデータベースを比較します ( 「比較 (Compare)」アイコンを選択して行います)。この詳しい手順については、35 ページの『CDT を使用したソース・データとターゲット・データの比較』を参照してください。
2. 「比較結果 (Comparison Results)」ツリーで、デプロイするエンティティを選択します。

利用可能なオプションは以下のとおりです。

- データベース全体の場合、 Sterling Selling and Fulfillment Foundation 「構成」アイコンを選択します。このオプションは、ステップ 1 でデータベース全体を比較した場合に限り、使用可能です。
- 特定の構成グループの場合、 「構成グループ (Configuration Group)」アイコンを選択します。
- 特定のドライバー・エンティティの場合、 「ドライバー・エンティティ (Driver Entity)」アイコンを選択します。
- ドライバー・テーブルの下にリストされている特定のレコードについては、そのレコードの  アイコンを選択します。

3. 「比較結果 (Comparison Results)」アクション・バーから、➡ 「デプロイ (Deploy)」アイコンを選択します。
  - デプロイメントが成功すると、「ステータス」パネルに成功のメッセージが表示され、データはターゲット・データベースにコミットされます。さらに、キャッシュは 21 ページの表 3 の説明に従って、httpurl フィールドで指定されたとおりに更新されます。
  - デプロイメントが失敗すると、「ステータス」パネルに解決すべきエラーが表示され、データはターゲット・データベースにコミットされません。
4. データを YFS\_RESOURCE テーブルからデプロイした場合、キャッシュをリフレッシュするために、ターゲット環境でアプリケーション・サーバーを再始動します。

## CDT によって生成される監査

構成データのデプロイメント中に、ターゲット・データベースに監査が生成されます。これらの監査は、変更のトラッキングやデータのロールバックに使用できます。これらの監査はソース・データベースからデプロイされるのではなく、<INSTALL\_DIR>/repository/entity ディレクトリーにあるエンティティ定義に基づいて生成されます。エンティティの監査参照の生成については、「Sterling Selling and Fulfillment Foundation データベースの拡張」を参照してください。

## CDT によって使用されるドライバー XML ファイル

Sterling Selling and Fulfillment Foundation は、データがデプロイされるテーブルの論理または機能グループを表示するために CDT によって使用される 2 つのドライバー xml ファイルを提供します。これらのファイルは、テーブル間の関数的依存関係に基づいて、テーブルを階層形式で編成します。実際のデータ・デプロイメントは、この構造または階層に従って行われます。config-db.xml および master-db.xml ファイルは、<INSTALL\_DIR>/database/cdt/ ディレクトリーにあります。

## コマンド・ライン・モードでの構成データのデプロイ このタスクについて

ユーザー・インターフェースによる対話を行ったり、ソースおよびターゲットの比較結果を表示したりせずに、構成データのデプロイメントを実行またはスケジュールしたいという状況が生じることがあります。

このことを実現するために、構成データをコマンド・ライン・モードでデプロイできます。データをコマンド・ライン・モードでデプロイすると、CDT は自動的にソースおよびターゲットの環境を比較して、構成データをデプロイします。

構成データをコマンド・ライン・モードでデプロイするには、以下のようになります。

### 手順

1. 47 ページの表 8 の説明に従って、Windows の場合は <INSTALL\_DIR>/bin/cdtshell.cmd ファイル (UNIX/Linux の場合は cdtshell.sh) でプロパティを設定します。

表 8. 構成デプロイメント・ツールのプロパティ

プロパティ	説明
SOURCE_DB	ソース・データベースの論理 ID。この値は、ydkprefs.xml ファイルの <SourceDatabases> エレメントの Name 設定に対応します。
SOURCE_PASSWORD	オプション。データのソースの宛先にデータベースを使用している場合、そのデータベース・インスタンスのパスワードを指定します。
TARGET_DB	ターゲット・データベースの論理 ID。この値は、ydkprefs.xml ファイルの <TargetDatabases> エレメントの Name 設定に対応します。
TARGET_PASSWORD	オプション。ターゲットの宛先にデータベースを使用している場合、そのデータベース・インスタンスのパスワードを指定します。
TARGET_HTTP_PASSWORD	サーバー上のキャッシュをリフレッシュするために必要な http ユーザーのパスワードを指定します。 注: このパラメーターは、サーバー上にリフレッシュするキャッシュがない場合には不要です。

- Windows の場合、<INSTALL\_DIR>/bin/setupfiles.cmd スクリプトを実行します (UNIX/Linux の場合は、setupfiles.sh)。
- Windows の場合、<INSTALL\_DIR>/bin/cdtshell.cmd スクリプトを実行します (UNIX/Linux の場合は、cdtshell.sh)。

## タスクの結果

このスクリプトを適切なタイミングで実行するようにスケジュールすることもできます。

CDT で次の MODE コマンド・ライン・オプションを使用できます。

- **MODE Deploy** — ソース・データベースからターゲット・データベースに変更をデプロイします。この MODE 引数とともに使用できるその他の各種オプションは、以下のとおりです。
  - **ExportDir <directory>** — 指定した <directory> が作成され、比較の結果はその <directory> に保管されます。
  - **ExportPassphrase <password>** — 指定したパスワードは、結果をエクスポートする際に、サポートされるインポートまたはエクスポート・データを暗号化するために使用されます。これは、ExportDir パラメーターが渡される場合にのみ適用できます。
  - **ImportDir <directory>** — 指定した <directory> に、エクスポートされた結果が含まれるようになります。ソース・データベースとターゲット・データベースを比較する代わりに、このエクスポートがロードされます。この引数を渡す場合、ソース・データベース・プロパティは使用されません。
  - **ImportPassphrase <password>** — 指定したパスワードは、インポート・ファイルからデータを暗号化解除するために使用されます。また、このパスワードは、エクスポート・ファイルを作成するために指定されるパスワードと一致している必要があります。

- DoNotSynchronize <YIN> — Y を渡す場合、比較のみ行われ、デプロイされるものはありません。デフォルトでは、エクスポートされた結果が自動的にデプロイされます。

例えば、比較を実行し、比較結果を特定のディレクトリーにエクスポートする場合に、変更をデプロイしたくない場合には、以下の引数を渡します。

```
-MODE Deploy -ExportDir C:\CDT\Reports -DoNotSynchronize Y
```

MODE java 引数を明示的に渡さない場合、CDT は MODE Deploy オプションをデフォルトとして使用します。

- MODE CDT2IE — CDT 比較エクスポート・フォーマットをインポートまたはエクスポート・ファイル・フォーマットに変換します。この MODE 引数とともに渡す必要がある 2 つの必須パラメーターは、以下のとおりです。

- InputFile <ydkexport.xml ファイルへのパス>
- OutputFile <作成するインポートまたはエクスポート・ファイルの名前>

以下に例を示します。

```
-MODE CDT2IE -InputFile C:\CDT\ydkexport.xml -OutputFile NewImportFile.xml
```

- MODE IE2CDT — インポートまたはエクスポート・ファイル・フォーマットを CDT 比較ファイルフォーマットに変換します。この MODE 引数とともに渡す必要がある 2 つの必須パラメーターは、以下のとおりです。

- InputFile <作成するインポートまたはエクスポート・ファイルの名前>
- OutputDir <CDT 比較ファイルを保管する必要があるディレクトリーへのパス>

以下に例を示します。

```
-MODE IE2CDT -InputFile MyImportFile.xml -OutputDir C:\CDT\Reports
```

- MODE LABELDEPLOY — 2 つのバージョン・ラベル間の監査変更をデプロイします。この MODE 引数とともに使用できる 2 つのオプションは、以下のとおりです。

- FromLabel <開始時のラベル ID>
- ToLabel <終了時のラベル ID>

以下に例を示します。

```
-MODE LABELDEPLOY -FromLabel MyLabel1 -ToLabel MyLabel2
```

---

## 第 10 章 構成データ・バージョン管理ツールの概要

---

### 構成データ・バージョン管理ツールの概要

構成データは、すべての実装環境にとって不可欠な部分です。多くの場合、実装の構成に対する変更を追跡する必要があります。さらに、構成データ内の変更が適切でないことが判明した場合、変更を元の状態にロールバックすることは非常に困難です。

オフサイトおよびオンサイトの実装モデルでは、マスター構成データは、実稼働環境がホストされているオンサイトで保守されます。フィックスバックを実稼働環境に適用する必要がある場合、オフサイトのテスト開発者は構成データの変更に関する指示をフィックスバック用に作成し、それをオンサイトの構成マネージャーに渡さなければなりません。つまり、ビジネス・ルールの特定の値を変更する必要があります。オンサイトの構成マネージャーは、構成変更を実稼働環境に複製する必要があります。

構成データのバージョンまたは構成データに対する一連の変更の追跡を容易にするために、Sterling Selling and Fulfillment Foundation には、構成データ・バージョン管理ツール (CDV Tool) が組み込まれています。これにより、ソース・データベースから変更を収集し、比較して、ターゲット・データベース (同じデータベースでも異なるデータベースでも可) にデプロイできます。

この機能を有効にするには、構成テーブルにおいて AuditRequired フラグが Y に設定され、かつテーブル名が config-db.xml に存在している必要があります。デフォルトでは、ほとんどの構成テーブルで、AuditRequired フラグが Y に設定されています。

CDV ツールは、構成テーブルのみをサポートします。

コンフィギュレーターでバージョン・ラベルを作成して、構成データで変更が生じた場合にタイム・スタンプを時系列で表すことができます。次に、システムは、バージョン・ラベルのタイム・スタンプ間で生じた構成データにおける変更を、システム内の監査情報に基づいて特定できます。

監査ページを構成する際に、YFS\_AUDIT テーブル・レコードをデプロイ可能なテーブルからパージすると、構成データ・バージョン管理ツールはそれらの変更をデプロイしません。

### 構成データ・バージョン管理ツールの機能

構成データ・バージョン管理 (CDV) ツールには、Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench (「WorkBench」として知られる) からアクセスできます。

構成データ・バージョン管理ツールを使用すると、ソース・データベースからさまざまなバージョン・ラベルを選択し、データを比較して、ターゲット・データベースに適用することができます。システムは、2 つのバージョン・ラベル間でソー

ス・データベースに対して行われた変更を識別します。それぞれの変更の詳細を調べて、競合を検出できます。競合の検査中に、これらの変更がターゲット・データベースに適用されます。すべての競合が解決されたら、変更をデプロイできます。

バージョン・ラベル・データは、YFS\_CONFIG\_VERSION\_LABEL テーブルに保管されます。config バージョン・ラベルの管理および検索には、manageConfigVersionLabel API および getConfigVersionLabel API が使用されます。

## 構成データ・バージョン・ラベルの例

以下の表は、VersionLabel\_1 および VersionLabel\_2 という 2 つのバージョン・ラベルと、それに関連する変更およびタイム・スタンプの例を示しています。

表9. バージョン・ラベルの例

バージョン・ラベル	監査済みの変更	タイム・スタンプ
	Org1 が作成されます。	5/1/2007 9:00
	Org2 が作成されます。	5/1/2007 9:10
VersionLabel_1		5/1/2007 9:20
	Org1 の DefaultFulfillmentType が「A」から「B」に更新されます。	5/1/2007 9:30
	Org1 の DefaultFulfillmentType が「B」から「C」に更新されます。	5/1/2007 9:40
	Org2 が削除されます。	5/1/2007 9:50
	Org3 が作成されます。	5/1/2007 10:00
VersionLabel_2		5/1/2007 10:10
	Org4 が作成されます。	5/1/2007 10:20
		現在のタイム・スタンプ

Sterling Development and Deployment WorkBench で、構成済みのバージョン・ラベルに基づいて変更を取得できます。これには、前方変更と後方変更の両方が含まれます。

### 前方変更

これは、以前のバージョン・ラベルからより新しいバージョン・ラベルまでの変更（または最新の変更）です。

このケースでは、VersionLabel\_1 から VersionLabel\_2 または Version Label\_1 から現在のタイム・スタンプまでの変更になります。これは、ユーザーがあるシステムから特定の変更を収集し、その変更を別のシステムにデプロイするというシナリオで使用できます。

### 前方変更の例

この例の基礎としてバージョン・ラベル・テーブルを使用します。ここで、ユーザーが VersionLabel\_1 から VersionLabel\_2 までの変更を取得する場合、以下の変更が収集され、後で別のシステムで適用されます。



表 10. 前方変更の例

適用される変更
Org1 の DefaultFulfillmentType を「A」から「C」に更新
Org2 を削除
Org3 を作成

## 後方変更

これは、現在時刻から以前のバージョン・ラベルまでの変更です。

監査データは、後で適用される変更としてリバース・ロジックで変換されます。

## 後方変更の例

この例の基礎としてバージョン・ラベル・テーブルを使用します。ここで、ユーザーが現行のタイム・スタンプから VersionLabel\_1 までのすべての変更を取得して、それらの変更を VersionLabel\_1 の後で行われたすべての変更のロールバックに使用できるようにする場合、以下の変更が取得されます。

表 11. 後方変更の例

ヘッダー
Org4 を削除
Org3 を削除
Org2 を作成
Org1 の DefaultFulfillmentType を「C」から「A」に更新

## CDV ツール・プリファレンスの設定について

構成データ・バージョン管理の競合を解決するためのプリファレンス (レポート・ディレクトリーなど) およびパラメーターを構成できます。

比較ツールを使用するには、その前にアプリケーション・マネージャーを使用してデータベースのバージョン・ラベルを作成する必要があります。構成バージョン・ラベルの作成に関する詳細は、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム 構成ガイド」を参照してください。

## 競合の処理

競合の処理は、プリファレンスの「バージョン・デプロイメントの競合の処理 (Conflict Handling for Version Deployment)」を選択すると可能になります。これは、ターゲット・データベース上で発生する、後続の変更をオーバーライドしないようにするために使用されます。

例えば、TaxPayerID が、ソース・データベース上の監査情報に基づいて、「A」から「B」に変更されたとします。しかし、ターゲット・データベース上では、TaxPayerID は「A」から「C」に変更されました。変更がソース・データベースから取得されて、ターゲット・データベースの現行値と比較されると、システムは値が「A」ではないことを検出し、確認のためにそのことをユーザーに示します。

3 つのレベルの適用方法があります。

- 古い値の検証 (Validate Old Values): これが有効にされている場合、システムは、ソース・データベースから取得した変更のレコードの属性の予期される古い値と、ターゲット・データベースでの対応するレコードの現行値の比較に基づいて競合を検出します。「詳細」画面に、競合する属性が表示されます。
  - ロック ID の検証 (Validate Lockid): これが有効にされている場合、システムは、ソース・データベースから取得した変更のレコードの予期されるロック ID と、ターゲット・データベースでの対応するレコードの現行ロック ID の比較に基づいて競合を検出します。競合が検出された場合、変更をデプロイできません。これが選択された場合、以下のようなことが起こります。
    - 「詳細」画面に、「ロック ID が競合しています - デプロイ・アクションは使用不可になります (Conflicting Lockid - Deploy Action is disabled)」というテキストが強調して表示されます。
    - 「予期される古い値 (Expected Old Values)」と競合する属性が表示されます。
    - デプロイ・アクションは、すべての競合が解決されるまで使用不可になります。
- Sterling Selling and Fulfillment Foundation リリース 8.0 以前に作成された監査レコードについては、ロック ID の検証は使用できません。
- 削除の前にレコードの存在を検証 (Validate Record Exists Before Delete): これが有効にされている場合、システムは、レコードの削除を試行する前に、ターゲット・データベース内でそのレコードの存在を検証します。レコードが存在しない場合、操作には競合というマークが付けられます。

## 構成データ・バージョン管理ツールのプリファレンスの設定 手順

1. 構成データ・バージョン管理ツールを始動します。構成データ・バージョン管理ツールの使用については、53 ページの『CDV ツールの実行』を参照してください。
2. 「ツール (Tools)」> 「構成データ・バージョンのデプロイメント (Configuration Data Version Deployment)」を選択します。
3. 「ログオン (Logon)」画面が表示されます。適切なソース・データベースおよびターゲット・データベースの名前を選択して、それぞれのパスワードを入力します。「OK」をクリックします。
4. 「比較 (Compare)」画面が表示されます。「プリファレンス」をクリックします。

「プリファレンス」ウィンドウで、構成データ・バージョン管理ツールのプリファレンスを定義できます。フィールド値の説明については、23 ページの表 4 を参照してください。

## CDV ツールの実行

### このタスクについて

Windows コンソールに、WorkBench 始動情報が表示されます。WorkBench の実行中は、コンソールを閉じないでください。コンソールを閉じると、ツールが閉じられ、作業は失われます。

構成データ・バージョン管理ツールを始動するには、以下のようになります。

### 手順

1. Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench を始動します。Microsoft Windows で、Windows の場合は ydk.cmd スクリプト (UNIX/Linux の場合は ydk.sh) を <INSTALL\_DIR>%bin ディレクトリーから実行します。
2. 「Sterling Selling and Fulfillment Foundation Development and Deployment WorkBench」メニューで、「ツール (Tools)」>「デプロイメント (Deployment)」>「構成データ・バージョンのデプロイメント (Configuration Data Version Deployment)」を選択します。これにより、構成データ・バージョン管理ツールの「ログオン (Logon)」ダイアログ・ボックスが開きます。

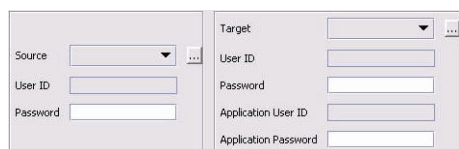


図 11. 構成データ・バージョン管理ツールのダイアログ・ボックス

3. 「ソース」ボタンを選択して、ソース・データベースについて適切な値を入力します。「ターゲット」ボタンを選択して、ターゲット・データベースについて適切な値を入力します。

完了したら、ダイアログ・ボックスを閉じます。指定された値は自動的に保存され、あるセッションから次のセッションまで持続します。

ソース・データベースまたはターゲット・データベースのいずれかの名前を変更すると、変換設定は失われます。以前の変換設定に戻すには、以前のソース・データベースおよびターゲット・データベースの名前に戻します。

「ソース・データベース (Source database)」ウィンドウと「ターゲット・データベース (Target database)」ウィンドウで、表 12 で説明されている、適用可能な値を指定します。

表 12. 構成デプロイメント・ツールの「ログオン (Logon)」ダイアログ・ボックス

フィールド	詳細
Name	論理データベース ID を指定します。ソースの場合、データのコピー元のデータベースを指定します。ターゲットの場合、データの書き込み先のデータベースを指定します。

表 12. 構成デプロイメント・ツールの「ログオン (Logon)」ダイアログ・ボックス (続き)

フィールド	詳細
className	<p>次のようにデータベース・ドライバーのクラス名を指定します。</p> <ul style="list-style-type: none"> <li>• Oracle を使用している場合、oracle.jdbc.OracleDriver に設定</li> <li>• Microsoft SQL Server 2005/2008 を使用している場合、com.microsoft.sqlserver.jdbc.SQLServerDriver に設定</li> <li>• DB2 を使用している場合、com.ibm.db2.jcc.DB2Driver に設定</li> </ul>
jdbcURL	<p>データベースに接続するための URL を指定します。</p> <ul style="list-style-type: none"> <li>• Oracle を使用している場合、jdbc:oracle:thin:@&lt;DatabaseServerHostname/IPAddress&gt;&lt;TNSListenerPortNumber&gt;:&lt;DatabaseSID&gt; に設定</li> <li>• Microsoft SQL Server 2005/2008 を使用している場合、jdbc:sqlserver://&lt;Database Server Hostname&gt;:&lt;Port Number&gt;;DatabaseName=&lt;Database name&gt; に設定</li> <li>• DB2 を使用している場合、jdbc:db2://&lt;Database Server Hostname&gt;:&lt;Port Number&gt;/&lt;Database name&gt; に設定</li> </ul>
dbType	<p>実行しているデータベースのタイプを指定します。次に示すように、すべて小文字で入力してください。</p> <ul style="list-style-type: none"> <li>• Oracle の場合、oracle と指定</li> <li>• Microsoft SQL Server の場合、sqlserver と指定</li> <li>• DB2 の場合、db2 と指定</li> <li>• XML データ・ソースの場合、xml と指定</li> </ul>
folder	<p>XML データ・ソースを使用している場合、XML ファイルのフォルダの場所の完全パスを指定します。</p>
httpurl	<p>ターゲット・データベースにのみ適用可能です。データがターゲット・データベースにデプロイされた後に、そのデータ・キャッシュがリフレッシュされるアプリケーション・サーバーの URL を指定します。使用する構文は次のとおりです。http://&lt;hostname/ip-address&gt;:&lt;port-number&gt;/&lt;Application&gt;/interop/InteropHttpServlet。ここで、&lt;hostname/IP-address&gt; はアプリケーションが実行されているサーバー・アドレス、&lt;port-number&gt; はアプリケーションが実行されているポート、&lt;Application&gt; はアプリケーションの名前 (例えば、smcfs) です。</p>
schema	<p>次のようにスキーマ所有者を指定します。</p> <ul style="list-style-type: none"> <li>• Oracle または DB2 データベースを使用しており、指定するユーザーが Sterling Selling and Fulfillment Foundation スキーマ所有者と異なる場合は、Sterling Selling and Fulfillment Foundation スキーマの所有者を指定します。</li> <li>• Microsoft SQL Server を使用している場合は、これをブランクのままにします。</li> </ul>
ユーザー	<p>データベースに関連付けられるユーザー名を指定します。</p>

4. 「ログオン (Logon)」ダイアログ・ボックスで、ユーザー名に関連付けられたパスワードを入力して、「OK」をクリックします。

5. 「比較 (Compare)」画面が表示されます。「開始バージョン・ラベル (From Version Label)」および「終了バージョン・ラベル (To Version Label)」を選択して、「OK」をクリックします。

システムは、2 つのデータベース・バージョンを比較します。比較が完了すると、図 12 に示す「比較結果 (Comparison Results)」画面が表示されます。

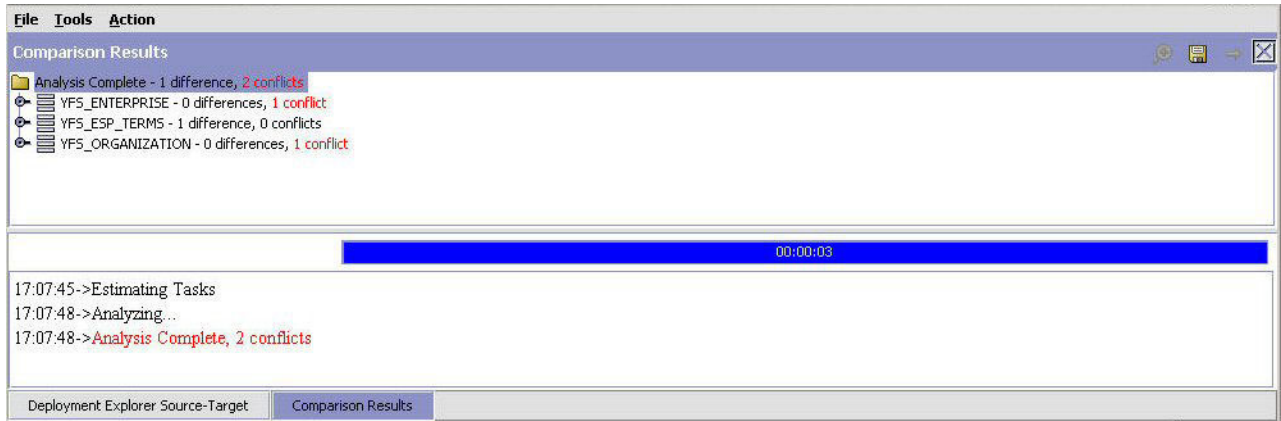


図 12. 「比較結果 (Comparison Results)」画面

この画面では、結果はテーブル名でグループ化されていることに注意してください。

6. 結果に含まれるレコードの詳細を表示するには、右クリックして、「詳細」を選択します。

アプリケーション・マネージャーを使用して、バージョン・ラベルを作成します。バージョン・ラベルの作成方法に関する詳細は、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム 構成ガイド」を参照してください。

Conflict Details			
YFS_ENTERPRISE		Record already exists	
		ENT-01[C-01]	
Name	Expected Old Value	Actual Old Value	New Value
		<input type="button" value="Previous"/> <input type="button" value="Next"/> <input type="button" value="Close"/>	
Changed			
Name	Old Value	New Value	
Unchanged			
Name	Value		
BillingaddressKey			
EnterpriseCode	ENT-01[C-01]		
UseDefaultOrderStatusRules	N		
Logoimagefile			

図 13. 「競合の詳細 (Conflict Details)」画面

表 13. 構成デプロイメント・バージョン管理ツールの「競合の詳細 (Conflict Details)」

フィールド	詳細
	最初の行には、競合を含むエンティティの名前 (例えば、YFS_ORGANIZATION)、競合のタイプ (例えば、「レコードが変更されました (Record has been modified)」)、およびユーザー・キー (例えば、DEFAULT) が表示されます。
名前	競合の原因となった値を持つ属性の名前を表示します。
予期される古い値 (Expected Old Value)	ターゲット・データベースで予期される古い値が表示されます。
実際の古い値 (Actual Old Value)	ターゲット・データベースに実際に存在する古い値を表示します。
新しい値	ソース・データベースでの変更が原因で取得される新しい値を表示します。
前へ	前のレコードを表示するには、このボタンをクリックします。
次へ	次のレコードを表示するには、このボタンをクリックします。
クローズ	「競合の詳細 (Conflict Details)」ウィンドウを閉じるには、このボタンをクリックします。
変更済み	変更された属性の名前を、その古い値および新しい値とともに表示します。
未変更	未変更の属性の名前を、その既存の値とともに表示します。

表示できる変更には、次の 3 つのタイプがあります。

- 変更: 「レコードの詳細 (Record Details)」画面に、変更済みの属性と未変更の属性のリストが表示されます。変更済みの属性の場合、古い値と新しい値が表示されます。(古い値は、ターゲット・データベース上の既存の値で、新しい値はソース・データベースから取得された変更の値です。)
- 挿入: 「レコードの詳細 (Record Details)」画面に、挿入される属性とその値のリストが表示されます。
- 削除: 「レコードの詳細 (Record Details)」画面に、エンティティの属性のリストが表示されます。


「前へ」および「次へ」ボタンを使用して、レコードをスクロールできます。レコードの詳細のレビューが完了したら、「クローズ」をクリックします。

## 構成データ・バージョン管理ツールからの結果のエクスポート このタスクについて

データベース内の競合を解決する前に、ファイルに対する変更を、別の環境に後でデプロイできるようにエクスポートする必要があります。これには「結果のエクスポート (Export Results)」オプションを使用します。調査のために結果をエクスポートすることは、競合が存在する場合のみ「必須」です。競合が存在しない場合、結果をエクスポートする必要はありません。

結果をファイルにエクスポートするには、以下のようにします。

### 手順

1. Sterling Development and Deployment WorkBench を始動します。
2. 「ツール (Tools)」 > 「構成データ・バージョンのデプロイメント (Configuration Data Version Deployment)」を選択します。
3. 「ログオン (Logon)」画面が表示されます。適切なソース・データベースおよびターゲット・データベースの名前を選択して、それぞれのパスワードを入力します。「OK」をクリックします。
4. 「比較 (Compare)」画面が表示されます。「開始バージョン・ラベル (From Version Label)」および「終了バージョン・ラベル (To Version Label)」を選択して、「OK」をクリックします。
5. 「比較結果 (Comparison Results)」アクション・バーから、 「デプロイ (Deploy)」アイコンを選択します。

### タスクの結果

このオプションは、以下を行います。

- 「プリファレンス」で構成された「レポート・ディレクトリー (Reports Directory)」の下に、  
export\_<FromVersionLabelID>\_<ToVersionLabelID>\_<CurrentTS> という命名規則に従って、新規のフォルダーを作成します。
- ydkversionexport.xml ファイルという名前のバージョン監査ファイルを作成します。

# 構成データ・バージョン管理ツールを使用したデータベースの競合の解決

## このタスクについて

データベースの競合は、バージョン監査ファイル `ydkversionexport.xml` を編集することによってのみ解決できます。`ydkversionexport.xml` ファイルには、システムに適用されるすべての変更が含まれます。検出されたすべての競合は、`<Conflict>` エlement内に記述されます。`<Conflict>` Elementには、競合のタイプや競合の説明などの情報が含まれます。

競合を解決するには、以下のようにします。

## 手順

1. 結果を `ydkversionexport.xml` ファイルにエクスポートします (まだ行っていない場合)。結果のエクスポートについて詳しくは、57 ページの『構成データ・バージョン管理ツールからの結果のエクスポート』を参照してください。
2. `ydkversionexport.xml` ファイルを開き、`<Conflict>` Elementを検索して、競合のタイプを検出します。CDV ツールは、以下のタイプの競合をサポートしません。

- `RECORD_ALREADY_EXISTS`
- `RECORD_DOES_NOT_EXIST`
- `RECORD_CHANGED`

以下は、`ydkversionexport.xml` ファイルの `<Conflict>` Element項目の例です。

```
<Conflict ConflictDescription="Record does not exist"
ConflictType="RECORD_DOES_NOT_EXIST"/>
```

## さまざまなタイプの競合を解決する方法

### `RECORD_ALREADY_EXISTS` タイプの競合の解決

このタイプの競合は、CDV が、データベースに既に存在するレコードに対して挿入操作を実行しようとする場合に生じます。このタイプの競合は、以下のいずれかの方法で解決できます。

- `<Audit>` Element全体を削除して、変更を抑制します。
- `<Audit>` Elementを変更し、挿入操作の代わりに更新操作にします。これは、`<ConflictAttribute>` Element内の `FoundOldValue` 属性を検索し、それを `<Audit>` Element内の `<OldValue>` で示されている値と一致するように設定することにより行います。その後、`<Audit>` Element内の `Operation` 属性の値を `Insert` から `Update` に変更します。

### `RECORD_DOES_NOT_EXIST` タイプの競合の解決

このタイプの競合は、CDV が、データベースに存在しないレコードに対して削除または更新操作を実行しようとする場合に生じます。このタイプの競合は、以下のいずれかの方法で解決できます。

- `<Audit>` Element全体を削除して、変更を抑制します。



- ターゲット・データベースで予期されるレコードを作成します。これは、Sterling Selling and Fulfillment Foundation UI または API を利用するか、または CDT を実行して行うことができます。

## RECORD\_CHANGED タイプの競合の解決

このタイプの競合は、CDV がデータベースで既に変更された値を持つレコードに対して更新操作を実行しようとする場合に生じます。このタイプの競合は、以下のいずれかの方法で解決できます。

- <Audit> エレメント全体を削除して、変更を抑制します。
- 値を変更したい場合は、OldValue 属性の値を、<ConflictAttribute> エレメント内の FoundOldValue 属性の値と一致するように作成または変更し、NewValue 属性の値を変更します。

以下は、ydkversionexport.xml ファイルの <Audit> エレメント項目の例です。

```
<Audit AuditKey="20081219134644594514" Operation="Insert"
  TableKey="20081219134644594512" TableName="YFS_QUEUE_SUBSCRIPTION">
  <AuditDetail AuditType="QueueSubscription">
    <IDs>
      <ID DataType="class java.lang.String" Name="QueueKey"
        Value="20081110160649488514"/>
    </IDs>
    <Attributes>
      <Attribute DataType="class com.yantra.yfc.date.YTimestamp"
        Name="Createts" NewValue="20081219134644"/>
      <Attribute DataType="class com.yantra.yfc.date.YTimestamp"
        Name="Modifys" NewValue="20081219134644"/>
      <Attribute DataType="class java.lang.String" Name="UserKey"
        NewValue="20051121111734438480"/>
    </Attributes>
  </AuditDetail>
  <Conflict ConflictDescription="Record already exists"
    ConflictType="RECORD_ALREADY_EXISTS">
    <ConflictAttribute ExpectedOldValue=""
      FoundOldValue="20081219134644594512"
      Name="QueueSubscriptionKey" NewValue="20081219134644594512"/>
    <ConflictAttribute ExpectedOldValue="" FoundOldValue=""
      Name="Createts" NewValue="2008-12-19T13:46:44+00:00"/>
    <ConflictAttribute ExpectedOldValue="" FoundOldValue=""
      Name="Modifys" NewValue="2008-12-19T13:46:44+00:00"/>
  </Conflict>
</Audit>
```

## CDV ツールによって生成されるバージョン監査ファイルのインポート

### このタスクについて

競合を解決した後、バージョン監査ファイル (ydkversionexport.xml) をインポートできます。

バージョン監査ファイルをインポートするには、以下のようにします。

### 手順

1. Sterling Development and Deployment WorkBench を始動します。
2. 「ツール (Tools)」 > 「デプロイメント (Deployment)」 > 「バージョン監査のインポート (Import Version Audits)」を選択します。

3. 「ログオン (Logon)」画面が表示されます。ターゲット・データベースの名前を選択して、そのターゲット・データベースのパスワードを入力します。バージョン監査ファイル (ydkversionexport.xml) を参照して、「OK」をクリックします。


## CDV ツールを使用したデータベース変更のデプロイ

### このタスクについて

あるデータベースから別のデータベースに変更をデプロイする前に、まず競合がないかどうか確認する必要があります。競合がある場合、競合を調査して解決するために、結果をファイルにエクスポートする必要があります。競合が解決されたら、バージョン監査ファイル (ydkversionexport.xml) を再インポートし、変更をターゲット・データベースにデプロイできます。競合の解決について詳しくは、58 ページの『構成データ・バージョン管理ツールを使用したデータベースの競合の解決』を参照してください。

データベース変更をデプロイするには、以下のようにします。

### 手順

1. Sterling Development and Deployment WorkBench を始動します。
2. 「ツール (Tools)」 > 「構成データ・バージョンのデプロイメント (Configuration Data Version Deployment)」を選択します。「ログオン (Logon)」画面が表示されます。
3. ソース・データベースおよびターゲット・データベースの名前を選択して、それぞれのパスワードを入力します。「OK」をクリックします。「比較 (Compare)」画面が表示されます。
4. 「開始バージョン・ラベル (From Version Label)」および「終了バージョン・ラベル (To Version Label)」を選択して、「OK」をクリックします。
5. 「比較結果 (Comparison Results)」アクション・バーから、 「デプロイ (Deploy)」アイコンを選択します。「入力」ウィンドウが表示されます。
6. 「このデプロイメントのラベルを入力してください (Enter a label for this deployment)」フィールドに、構成データのデプロイ元のラベル番号を入力します。
  - デプロイメントが成功すると、「ステータス」パネルに成功のメッセージが表示されます。データはターゲット・データベースにコミットされ、キャッシュは 21 ページの表 3 の説明に従って、httpurl フィールドで指定されたとおりに更新されます。
  - デプロイメントが失敗すると、「ステータス」パネルに解決すべきエラーが表示され、データはターゲット・データベースにコミットされません。
  - マルチスキーマ環境では、構成データをデプロイした後、必ずコロニー・マップ・シンクロナイザー・エージェントを実行します。


## CDV ツールを使用した変更のロールバック

### このタスクについて

指定したバージョン・ラベルに対する変更をロールバックできます。CDV ツールは、ラベルから現在のタイム・スタンプまですべてのレコードを監査し、変更を見つけ、それらの変更を取り消して、それをデータベースに適用し直します。

特定のラベルに対する変更をロールバックするには、以下のようになります。

### 手順

1. Sterling Development and Deployment WorkBench を始動します。
2. 「ツール (Tools)」 > 「デプロイメント (Deployment)」 > 「バージョン・ラベルへのデータのロールバック (Rollback Data To Version Label)」を選択します。
3. データベースの名前を選択し、パスワードを入力します。「OK」をクリックします。
4. 「開始バージョン・ラベル (To Version Label)」を選択して、「OK」をクリックします。「比較 (Compare)」画面が表示されます。
5. 「比較結果 (Comparison Results)」アクション・バーから、 「デプロイ (Deploy)」アイコンを選択します。「入力」ウィンドウが表示されます。
6. 「このデプロイメントのラベルを入力してください (Enter a label for this deployment)」フィールドに、構成データのデプロイ元のラベル番号を入力します。
  - デプロイメントが成功すると、「ステータス」パネルに成功のメッセージが表示されます。データはターゲット・データベースにコミットされ、キャッシュは 21 ページの表 3 の説明に従って、httpurl フィールドで指定されたとおりに更新されます。
  - デプロイメントが失敗すると、「ステータス」パネルに解決すべきエラーが表示され、データはターゲット・データベースにコミットされません。



---

## 第 11 章 構成デプロイメント・ツールのトラブルシューティング

---

### CDT のトラブルシューティング

操作中に、構成デプロイメント・ツールは、各操作のステータスを把握できるようにするメッセージを「ステータス」パネルに表示します。これらのメッセージは、以下のように分類できます。

- ステータス (Status)
- 警告
- 予期しないエラー

『構成デプロイメント・ツールのメッセージ』には、表示される可能性がある各種のメッセージと、実行する必要がある関連修正処置が説明されています。

---

### 構成デプロイメント・ツールのメッセージ

構成デプロイメント・ツールの使用時に、情報メッセージまたは警告メッセージのいずれかが表示される場合があります。これらのメッセージ・タイプの説明は以下のとおりです。

#### 情報メッセージ

情報メッセージは、実行中の操作の状況を表します。これらのメッセージは、「ステータス」パネルにデフォルトの色 (通常は黒) で表示されます。情報メッセージの例としては、以下があります。

- データベース・キャッシュをリフレッシュしています (Refreshing database cache)
- デプロイメント操作が開始しました (Deployment operation started)
- テーブル YFS\_ORGANIZATION の読み取り中です (Reading table YFS\_ORGANIZATION)

#### 警告メッセージ

警告メッセージには通常、修正処置が必要です。警告メッセージは、「ステータス」パネルに赤で表示されます。CDT は、このセクションで説明されている警告メッセージを生成することがあります。

**警告 - テーブル <name> から <name2> に対する FK チェックが失敗しました (WARNING - FK check failed for table <name> to <name2>)**

この警告メッセージは通常、デプロイしようとしている構成データと、ターゲット・データベースのデータとの間に不整合があることを示します。

この問題を分析し、修正するには、以下のようにします。

1. デプロイしているデータ・セットのサイズを判別します。このエラーは通常、非常に小さいデータ・セット（ドライバー・エンティティーのみや構成グループのみなど）をデプロイしようとしている場合に発生します。例えば、パイプラインをデプロイしているときに、そのパイプラインが属する文書タイプがデプロイメントで選択されていない場合、このエラーが発生します。

このエラーを解決するには、より大きいセットをデプロイすることを選択します。例えば、レコードをデプロイする代わりに、可能であればグループ全体をデプロイします。

2. あるグループについてこのエラーが依然として発生するか、または特定のレコードのみデプロイする必要がある場合、データをデプロイする前に、外部テーブルを同期します。
3. 場合によっては、ソース・データベースのデータの不整合により、このエラーが発生します。そのような場合、続行する前に、不整合の原因を修正する必要があります。

### **警告 - キャッシュのリフレッシュが失敗しました (WARNING - Cache Refresh Failed)**

このエラーは、CDT がターゲット環境にあるアプリケーション・サーバー・クラスターに、新しくデプロイされた構成変更について通知できなかったことを示します。キャッシュのリフレッシュが失敗した理由が「ステータス」パネルに表示されます。

この問題を分析し、修正するには、以下のようになります。

1. ターゲット・データベースの `httpurl` フィールドに指定された URL を確認します。 `httpurl` には「ログオン (Logon)」ダイアログ・ボックスからアクセスできます。 `httpurl` が、以下のフォーマットで、アプリケーション・サーバーの実行中のインスタンスを指していることを確認します。

```
http://<hostname/ip-address>:<port-number>/smcfs/interop/
InteropHttpServlet。ここで、hostname、ip-address、および port-number は、アプリケーション・サーバーへの接続に使用されるパラメーターです。
```

2. ターゲット環境が実行中でない場合、アクションは不要です。 Sterling Selling and Fulfillment Foundation は、始動時に最新の構成データを自動的に読み取りません。
3. ターゲット環境が実行中の場合、 Sterling Selling and Fulfillment Foundation システム管理コンソールを使用して、不整合のあるデータベース・キャッシュを手動で除去する必要があります。このステップを実行しないと、 Sterling Selling and Fulfillment Foundation が、変更された構成を認識しない場合があります。

### **警告 - プログラムはターゲット・データベースでいくつかの中止レコードを検出しました (WARNING - The program detected a few abandoned records in the target database.)**

ほとんどの場合、中止レコードは無害であり、 Sterling Selling and Fulfillment Foundation の誤操作につながることはありません。デフォルトでは、CDT は中止レコードをそのままにします。

この警告は通常、以下の状況の結果として発生します。詳細については、7 ページの『外部で管理されている構成データのインポート』で説明されています。

- CDT が、レコードが有効なドライバー・エンティティに属していないと判断した場合 (例えば、既に存在していないプロセス・タイプのパイプライン)。
- CDT が、すべての従属テーブルを無視するのではなく、特定のテーブルを無視するように構成されている場合。

この問題を分析し、修正するには、以下のようにします。

1. `-DShowAbandoned=Y` Java パラメーターを `ydk.cmd` スクリプトに追加します。
2. `ydk.cmd` スクリプトを実行します。CDT は、中止レコードを検出すると、「中止レコード」というグループを動的に作成し、「比較結果 (Comparison Results)」ウィンドウに表示します。
3. これらのレコードを検証し、無視するか、またはターゲットから削除します。

---

## 構成デプロイメント・ツールでの予期しないエラー・メッセージ

重大度に応じて、予期しないエラーに関するメッセージが、以下のいずれかの場所に表示されます。

- 「CDT ステータス」パネル (赤で表示)
- 構成デプロイメント・ツールの起動に使用される Microsoft Windows コンソール

これらのエラーを分析し、修正するには、以下のようにします。

- エラーがメモリー不足状態を示す場合、より小さいデータ・セットで以前の操作を試行します。
- システムの仕様が、「*Sterling Selling and Fulfillment Foundation: システム要件ガイド*」で説明されている推奨事項に準拠していることを確認します。
- 構成デプロイメント・ツールで使用できるメモリーを増やすには、`<INSTALL_DIR>/bin/tmp.cmd` ファイルを編集し、`-ms` 引数の値をより大きい値に設定します。デフォルトでは、`tmp.cmd` ファイル内で、`HEAP_FLAGS` Java パラメーターは次のように設定されています。

```
set HEAP_FLAGS=-Xms768m -Xmx768m
```

例えば、構成全体を比較する場合、一度に 1 つのグループを比較するようにしてください。同様のことが、デプロイメント操作にも当てはまります。

その他の場合、根本的なエラーおよび詳細トレースが表示されます。これは、インストールが未完了であるか、インストールに障害があること、または指定したランタイム・パラメーターが正しくないことを示す可能性があります。

---

## cdtshell スクリプトで構成データをエクスポートする場合の例外

CDT を使用して、データベースが `SOURCE_DB`、XML ファイルが `TARGET_DB` である構成データをエクスポートする場合、`cdtshell.cmd` (または `.sh`) スクリプトは、`java.lang.StringIndexOutOfBoundsException` をスローします。

この例外を分析し、修正するには、以下のようにします。

ExportDir と、XML ファイルのフォルダーの場所が同じではないことを確認します。

---

## データ・ロールバック・スクリプト

データをステージング環境から実稼働環境にデプロイする前に、実動構成データのスナップショットを取ることが推奨されています。このスナップショットにより、障害発生時にデプロイメント操作のロールバックを実行できます。Sterling Selling and Fulfillment Foundation は、以下のロールバック・スクリプトを提供しています。

- バックアップ・スクリプト - Sterling Selling and Fulfillment Foundation 構成データからのデータを含む、複数のファイルを作成します。
- リストア・スクリプト - バックアップ・スクリプトによって作成されたファイルを使用して、Sterling Selling and Fulfillment Foundation の構成を以前に良好と認識されていた状態にリストアします。

<INSTALL\_DIR>/bin/backupScriptGen.xml スクリプトは、バックアップおよびリストア・スクリプトを生成します。データ・ロールバック・スクリプトの実行について詳しくは、「*Sterling Selling and Fulfillment Foundation インストール・ガイド*」を参照してください。



---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*IBM Corporation*

*J46A/G4*

*555 Bailey Avenue*

*San Jose, CA 95141-1003*

*U.S.A.*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、**IBM** 所定のプログラム契約の契約条項、**IBM** プログラムのご使用条件、またはそれと同等の条項に基づいて、**IBM** より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのもと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

**IBM** 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。**IBM** は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。**IBM** 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

**IBM** の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている **IBM** の価格は **IBM** が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© IBM 2011. このコードの一部は、IBM Corp. のサンプル・プログラムの派生物です。© Copyright IBM Corp. 2011.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com)<sup>®</sup> は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、および PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は、英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

Intel、Intel (ロゴ)、Intel Inside、Intel Inside (ロゴ)、Intel Centrino、Intel Centrino (ロゴ)、Celeron、Intel Xeon、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Cell Broadband Engine, Cell/B.E は、米国およびその他の国における Sony Computer Entertainment, Inc. の商標であり、同社の許諾を受けて使用しています。

Linear Tape-Open, LTO, LTO ロゴ、Ultrium および Ultrium ロゴは、米国およびその他の国における HP、IBM Corp. および Quantum の商標です。

Connect Control Center<sup>®</sup>、Connect:Direct<sup>®</sup>、Connect:Enterprise<sup>™</sup>、Gentran<sup>®</sup>、Gentran<sup>®</sup>:Basic<sup>®</sup>、Gentran:Control<sup>®</sup>、Gentran:Director<sup>®</sup>、Gentran:Plus<sup>®</sup>、Gentran:Realtime<sup>®</sup>、Gentran:Server<sup>®</sup>、Gentran:Viewpoint<sup>®</sup>、Sterling Commerce<sup>™</sup>、Sterling Information Broker<sup>®</sup>、および Sterling Integrator<sup>®</sup> は、Sterling Commerce<sup>™</sup>、Inc.、IBM Company の商標です。

# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [カ行]

構成データのデプロイ  
  cdtshell.cmd/sh スクリプトでエクスポートする場合の例外 65  
構成データのロールバック 66  
構成データ・バージョン管理 (CDV) ツール 49  
構成デプロイメント・ツール 1  
  インポート  
    構成の差異 42  
  エクスポート  
    比較結果 40  
  および外部で管理されているデータ追加 8  
    ベスト・プラクティス 8  
    無視 8  
  外部キーの相互依存関係 9  
    トラブルシューティング 64  
  カスタム・テーブル指定 23  
  グループ  
    構成グループ 7  
    ドライバー・エンティティ 7  
計画  
  アップグレード 17  
  システム要件 15  
  セキュリティ 16  
  前提条件 16  
  変更管理戦略 16  
タスク  
  エレメントの変換 34  
  カスタム・クラスの指定 23  
  コマンド・ライン・モードでのデータのデプロイ 46  
  差異のレポートの生成 41  
  追加するテーブルの指定 25, 32  
  データベース間の差異の検証 38  
  変換の指定 24  
  無視するテーブルの指定 25, 32  
  レポート・ディレクトリーの指定 23, 30  
  ロールバックの実行 66  
データ変換 9, 33  
  指定 24

構成デプロイメント・ツール (続き)  
  トラブルシューティング  
    キャッシュのリフレッシュ・エラー 64  
  中止レコード・エラー 64  
  予期しないエラー 65  
  ロールバック・スクリプト 66  
  FK チェック・エラー 63  
パフォーマンス  
  時間の見積もり 15  
ユーザー・インターフェース  
  「ステータス」パネル 13  
  「比較結果 (Comparison Results)」ウィンドウ 12  
  Deployment Explorer ウィンドウ 11  
コロニー 38  
  「コロニー ID」 38

## [サ行]

実稼働環境 3  
ステージング環境 3  
組織によって決定される構成 4, 21  
組織によって決定されるマスター・データ 4, 21

## [タ行]

ドライバー XML ファイル 46  
ドライバー・エンティティ 7

## [マ行]

マルチスキーマ  
  構成データ 21  
  コロニー 38  
  ソースおよびターゲット環境 3  
  データの比較 37, 38  
メタデータ 3, 21

## C

CDT (構成デプロイメント・ツール) 1

## H

HUB データ 3, 21







Printed in Japan