



Copyright © 2013 NTT DATA INTRAMART
CORPORATION

目次

- 改訂情報
- はじめに
 - 本書の目的
 - 前提条件
 - 対象読者
- 各種インストール・設定変更
- intra-mart Accel Platform 構成ファイルの作成
- WebSphereの設定
 - Java VM引数の設定
 - トランザクション・タイムアウトの設定
 - データベース接続の設定
 - warファイルのデプロイ
 - クラスローダーの設定
 - Webアプリケーションの開始
- テナント環境の構築
- セットアップで困ったら・・・
- アップデート・パッチの適用

改訂情報

変更年月日	変更内容
2013-04-01	初版
2013-07-01	第3版 下記を追加・変更しました ■ 「Java VM引数の設定」を追加 ■ 「タイムゾーンの設定」を追加
2013-10-01	第3版 下記を追加・変更しました ■ 「Java VM引数の設定」を修正 ■ 「データソースの設定」を修正
2014-01-01	第4版 下記を追加・変更しました ■ 目次に「アップデート・パッチの適用」へのリンクを追加 ■ 「PostgreSQLの設定」を修正
2015-04-01	第5版 下記を追加・変更しました ■ 目次に「Java VM引数の設定」へのリンクを追加
2015-12-01	第6版 下記を追加・変更しました ■ 「Java VM引数の設定」を修正

はじめに

本書の目的

本書では WebSphere Application Server 8.5.5 に intra-mart Accel Platform のセットアップを行う手順について説明します。

前提条件

リリースノートに記載されているシステム要件を満たしている必要があります。

詳細は「[リリースノート](#)」を参照してください。

対象読者

以下の利用者を対象としています。

- WebSphere Application Server 8.5.5 に intra-mart Accel Platform のセットアップを行われる方

intra-mart Accel Platform のセットアップに必要なコンポーネントのインストールおよび設定を行います。

具体的な手順は「[intra-mart Accel Platform セットアップガイド](#)」の「[intra-mart Accel Platform を利用するためのミドルウェアのインストールと設定](#)」を参照してください。

intra-mart Accel Platform の設定およびwarファイルの出力を行います。

具体的な手順は「[intra-mart Accel Platform セットアップガイド](#)」の「[WARファイルの作成](#)」を参照してください。

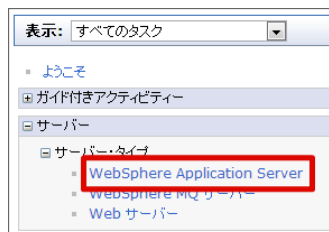
WebSphereの設定を行います。

WebSphere Application Server 8.5.5 のインストール手順については WebSphere Application Server 8.5.5 のマニュアルを参照してください。

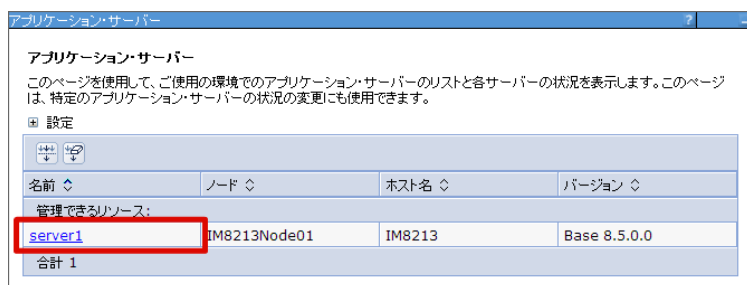
Java VM引数の設定

Java VM 引数の設定を行います。

1. メニューから[サーバー]-[サーバー・タイプ]-[WebSphere Application Server] を選択します。



2. サーバーの名前を選択します。



3. [サーバー・インフラストラクチャー]-[Java およびプロセス管理]-[プロセス定義] を選択します。



4. [追加プロパティ]-[Java 仮想マシン] を選択します。

5. ヒープサイズを入力します。

6. 汎用 JVM 引数を入力します。ここでは下記の引数を設定します。

-XX:PermSize	Permanent領域の初期値
-XX:MaxPermSize	Permanent領域の最大値
-Duser.timezone	Javaランタイム システムプロパティのタイムゾーン
-Dfile.encoding	Javaランタイム システムプロパティのファイルエンコーディング
-Xgcpolicy:gencon	ガーベッジ・コレクション・ポリシー

下記ではPermanent領域の初期値に512MB、最大値に512MBを設定しています。

```
-XX:PermSize=512m -XX:MaxPermSize=512m -Duser.timezone=UTC -Dfile.encoding=UTF-8 -Xgcpolicy:gencon
```



コラム

タイムゾーンのデフォルト値は、Javaランタイム に依存します。



注意

タイムゾーンは、「設定ファイルリファレンス タイムゾーンマスタ」に存在するものを設定してください。

**注意**

タイムゾーンの設定は運用開始前に行ってください。
運用開始後に変更した場合、日時データに不整合が発生します。

**注意**

ガーベッジ・コレクション・ポリシーの設定はご利用の環境に合わせて設定を行ってください。
ガーベッジ・コレクション・ポリシーについての詳細は以下をご覧ください。

- ケース・スタディー: WebSphere Application Server V7、V8 のパフォーマンス・チューニング

7. [JIT を使用不可にする] のチェックボックスにチェックされていない事を確認し、[OK] ボタンをクリックします。

**注意**

JITを有効にしている場合、正常に動作しない可能性があります。この場合はJITを無効にしてください。
過去の事例として、スクリプト開発モデルで想定する型と違う型で処理されるなどの問題が確認されています。
これらの現象は環境等に依存する場合もあり、事前に十分な検証を行い、JITの利用有無を判断してください。

8. [保存]をクリックします。これで、Java VM 引数の設定は終わりです。

9. WebSphere を再起動します。

トランザクション・タイムアウトの設定

トランザクション・タイムアウトの設定を行います。

テナント環境構築時などの時間がかかるトランザクション処理のために、トランザクション・タイムアウトの時間を変更します。

1. メニューから[サーバー]-[サーバー・タイプ]-[WebSphere Application Server] を選択します。

2. サーバーの名前を選択します。

アプリケーション・サーバー

アプリケーション・サーバー

このページを使用して、ご使用の環境でのアプリケーション・サーバーのリストと各サーバーの状況を表示します。このページは、特定のアプリケーション・サーバーの状況の変更にも使用できます。

設定

名前	ノード	ホスト名	バージョン
server1	IM8213Node01	IM8213	Base 8.5.0.0
合計	1		

3. [コンテナー・サービス]-[トランザクション・サービス] を選択します。

アプリケーション・サーバー

アプリケーション・サーバー > server1

このページを使用して、アプリケーション・サーバーを構成します。アプリケーション・サーバーとは、エンタープライズ・アプリケーションを実行するのに必要なサービスを提供するサーバーです。

ランタイム 構成

一般プロパティ

名前: server1

ノード名: IM8213Node01

☐ 開発モードでの実行

☒ 並列始動

☐ 必要に応じてコンポーネントを開始

内部サーバー・クラスへのアクセス: 許可

サーバー固有のアプリケーション設定

クラス・ローダー・ポリシー: 複数

クラス・ロード・モード: 最初に親クラス・ローダーをロードしたクラス

コンテナー設定

- セッション管理
- SIP コンテナー設定
- Web コンテナー設定
- ポートレット・コンテナー設定
- EJB コンテナー設定
- コンテナー・サービス
 - アプリケーション・プロファイル・サービス
 - トランザクション・サービス
 - 動的キックオフ・サービス
 - 補正サービス
 - Internationalization Service
 - デフォルト Java Persistence API 設定
 - オブジェクト・プール・サービス
 - ORB サービス
 - 始動 Bean サービス

4. [合計トランザクション存続時間タイムアウト] を変更し、[OK] ボタンをクリックします。

アプリケーション・サーバー

アプリケーション・サーバー > server1 > トランザクション・サービス

このページを使用して、トランザクション・サービスの設定を指定します。トランザクション・サービスは、複数のリソース・マネージャーに対する更新を調整して、データ・トランザクションの原子性更新がアプリケーションまたはそのアプリケーションがデプロイされているコンテナーによって始動および終了されることを保証できる、サーバーのランタイム・コンポーネントです。

ランタイム 構成

一般プロパティ

トランザクション・ログ・ディレクトリ

* 合計トランザクション存続時間タイムアウト: 300 秒間

* Async 応答タイムアウト: 30 秒間

* クライアント非活動タイムアウト: 60 秒間

* 最大トランザクション・タイムアウト: ...

ヒューリスティックな再試行待ち: 0 秒間

☐ ヒューリスティックレポート作成のためにロギングを使用可能にする

ヒューリスティックな完了指示: ロールバック

☐ ヒューリスティック障害を容認

☒ ファイル・ロックを使用可能にする

☒ トランザクション調整許可を使用可能にする

デフォルト t WS-Transaction 仕様レベル: 1.0

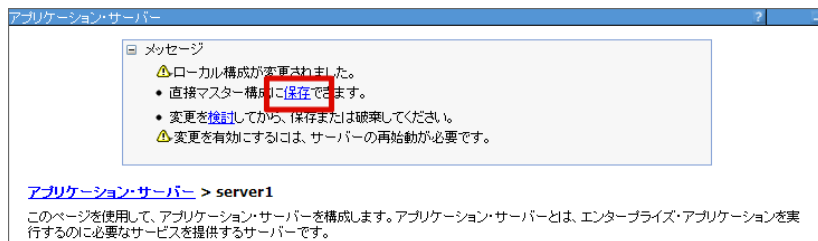
外部 WS-Transaction HTTP(S) URL 接頭部

接頭部の選択:

カスタム接頭部の指定:

適用 OK リセット キャンセル

5. [保存]をクリックします。これで、トランザクション・タイムアウトの設定は終わりです。



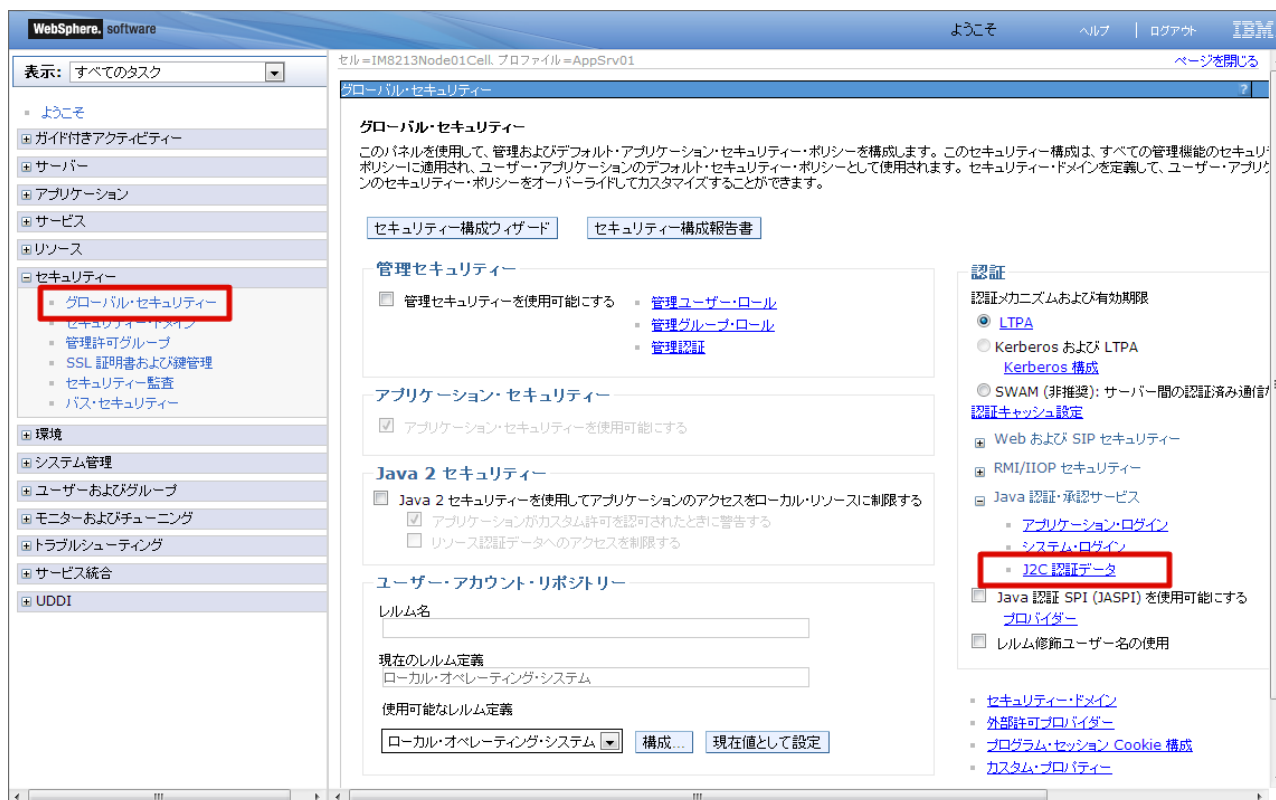
データベース接続の設定

データベース接続の設定を行います。

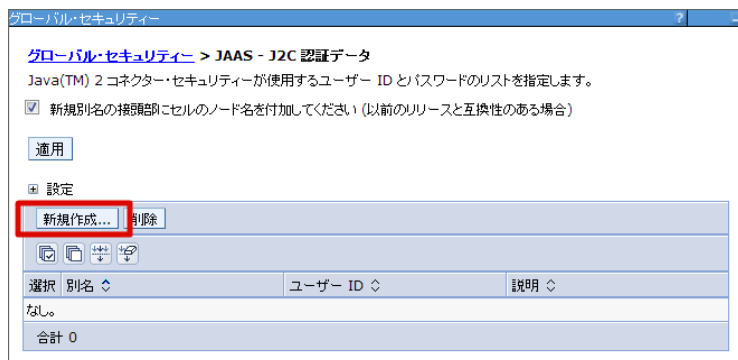
DBユーザ情報の設定

DBユーザ情報の設定を行います。

- WebSphereの管理コンソールにログインし、[セキュリティ]-[グローバル・セキュリティ]の[認証]-[Java 認証・承認サービス]-[J2C 認証データ]を選択します。



- [新規作成] ボタンをクリックします。



- 別名 (任意)、DBユーザID、パスワードを入力し、[OK] ボタンをクリックします。
ここでは別名を「db_user」とします。

グローバルセキュリティ > JAAS - J2C 認証データ > 新規作成...

Java(TM) 2 コネクタ・セキュリティが使用するユーザー ID とパスワードのリストを指定します。

一般プロパティ

* 別名
db_user

* ユーザー ID
imart

* パスワード

説明

適用 OK リセット キャンセル

4. [保存]をクリックします。これで、DBユーザ情報の設定は終わりです。

グローバルセキュリティ > JAAS - J2C 認証データ

Java(TM) 2 コネクタ・セキュリティが使用するユーザー ID とパスワードのリストを指定します。

☒ 新規別名の接頭語にセルのノード名を付加してください (以前のリリースと互換性のある場合)

適用

設定

新規作成... 削除

選択 別名 ユーザー ID 説明

管理できるリソース:	別名	ユーザー ID	説明
☐	IM8213Node01/db_user	imart	

合計 1

JDBCプロバイダの設定

JDBCプロバイダの設定を行います。使用するデータベースに応じて設定を行ってください。

Oracleの設定

Oracleを使用する場合の設定を行います。

1. メニューから[リソース]-[JDBC]-[JDBC プロバイダー]を選択します。

表示: すべてのタスク

- ようこそ
- ガイド付きアクティビティ
- サーバー
- アプリケーション
- サービス
- リソース
 - スケジューラ
 - オブジェクト・プール・マネージャ
 - JMS
 - JDBC
 - JDBC プロバイダー
 - データソース
 - データソース (WebSphere Application Server V4)
 - リソース・アダプター
 - 非同期 Bean
 - キャッシュ・インスタンス
 - メール
 - URL
 - リソース環境

2. 「JDBC プロバイダー」画面で [新規作成] ボタンをクリックします。

JDBC プロバイダー

このページを使用して、JDBC プロバイダーのプロパティを編集します。JDBC プロバイダー・オブジェクトは、ご使用の環境の特定のベンダー・データベースにアクセスするための特定の JDBC ドライバー実装クラスをカプセル化します。このタスクの詳細については、[ガイド付きアクティビティ](#)で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=IM8213Node01, サーバー=server1

設定

新規作成... 削除

選択 名前 有効範囲 説明

管理できるリソース:

☐	Derby JDBC Provider	ノード=IM8213Node01,サーバー=server1	Derby embedded non-XA JDBC Provider
--------------------------	-------------------------------------	-------------------------------	-------------------------------------

合計 1

3. [データベース・タイプ]、[プロバイダー・タイプ]、[実装タイプ]を設定し、[次へ]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

→ ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラスパス情報を入力

ステップ 3: 要約

データベースのアクセスに必要な固有のベンダー JDBC ドライバー実装クラスをカプセル化する、JDBC プロバイダーの基本構成値

有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* データベース・タイプ
Oracle

* プロバイダー・タイプ
Oracle JDBC Driver

* 実装タイプ
接続プール・データ・ソース

* 名前
Oracle JDBC Driver

説明
Oracle JDBC Driver

次へ キャンセル

設定項目	値
データベース・タイプ	Oracle
プロバイダー・タイプ	Oracle JDBC Driver
実装タイプ	接続プール・データ・ソース

4. JDBCドライバのディレクトリ・ロケーションを設定し、[次へ]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

→ ステップ 2: データベース・クラスパス情報を入力

データベース・クラスパス情報を入力

JDBC プロバイダーの定義に WebSphere(R) Application Server が使用する、JDBC ドライバー・クラス・ファイルのクラスパス・エンタリイは、ENTER (実行) キーを使用して区切ります。パス区切り文字 (「;」や「:」など) は使用できません。値が表示されている。

クラスパス:
\${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

適用

WebSphere 変数 \${ORACLE_JDBC_DRIVER_PATH} として保存される "ojdbc6.jar" のディレクトリ・ロケーション
c:/jdbc_driver

前へ **次へ** キャンセル

5. 設定内容を確認し、[終了]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラスパス情報を入力

→ ステップ 3: 要約

要約

アクションの要約:

オプション	値
有効範囲	cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1
JDBC プロバイダー名	Oracle JDBC Driver
説明	Oracle JDBC Driver
クラスパス	\${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
\${ORACLE_JDBC_DRIVER_PATH}	c:/jdbc_driver
実装クラス名	oracle.jdbc.pool.OracleConnectionPoolDataSource

前へ 終了 キャンセル

6. [保存]をクリックします。これで、JDBCプロバイダーの設定は終わりです。

JDBC プロバイダー

メッセージ

- ローカル構成が変更されました。
- 直接マスタ構成に保存できます。
- 変更を検討してから、保存または破棄してください。
- 変更を有効にするには、サーバーの再始動が必要です。

JDBC プロバイダー

このページを使用して、JDBC プロバイダーのプロパティを編集します。JDBC プロバイダー・オブジェクトは、ご使用の環境の特定のベンダー・データベースにアクセスするための特定の JDBC ドライバー実装クラスをカプセル化します。このタスクの詳細については、[ガイド付きアクティビティ](#)で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=IM8213Node01Cell, ノード=IM8213Node01, サーバー=server1

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=IM8213Node01, サーバー=server1

設定

新規作成... 削除

選択 名前 有効範囲 説明

管理できるリソース:

☐	Derby JDBC Provider	ノード=IM8213Node01,サーバー=server1	Derby embedded non-XA JDBC Provider
☐	Oracle JDBC Driver	ノード=IM8213Node01,サーバー=server1	Oracle JDBC Driver

合計 2

DB2の設定

DB2の設定を行います。

1. メニューから[リソース]-[JDBC]-[JDBC プロバイダー]を選択します。

表示: すべてのタスク

- ようこそ
- ガイド付きアクティビティ
- サーバー
- アプリケーション
- サービス
- リソース
 - スケジューラー
 - オブジェクト・プール・マネージャ
 - JMS
 - JDBC
 - JDBC プロバイダー
 - データソース (WebSphere Application Server V4)
 - リソース・アダプター
 - 非同期 Bean
 - キャッシュ・インスタンス
 - メール
 - URL
 - リソース環境

2. 「JDBC プロバイダー」画面で [新規作成] ボタンをクリックします。

JDBC プロバイダー

このページを使用して、JDBC プロバイダーのプロパティを編集します。JDBC プロバイダー・オブジェクトは、ご使用の環境の特定のベンダー・データベースにアクセスするための特定の JDBC ドライバー実装クラスをカプセル化します。このタスクの詳細については、[ガイド付きアクティビティ](#)で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=IM8213Node01, サーバー=server1

設定

新規作成... 削除

選択 名前 有効範囲 説明

管理できるリソース:

☐	Derby JDBC Provider	ノード=IM8213Node01,サーバー=server1	Derby embedded non-XA JDBC Provider
--------------------------	-------------------------------------	-------------------------------	-------------------------------------

合計 1

3. [データベース・タイプ]、[プロバイダー・タイプ]、[実装タイプ]を設定し、[次へ]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

→ ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラス・パス情報を入力

ステップ 3: 要約

データベースのアクセスに必要な固有のベンダー JDBC ドライバー実装クラスをカプセル化する、JDBC プロバイダーの基本構成値

有効範囲

cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* データベース・タイプ

DB2

* プロバイダー・タイプ

DB2 Universal JDBC Driver Provider

* 実装タイプ

接続プール・データ・ソース

* 名前

DB2 Universal JDBC Driver Provider

説明

One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.

次へ キャンセル

設定項目	値
データベース・タイプ	DB2
プロバイダー・タイプ	DB2 Universal JDBC Driver Provider
実装タイプ	接続プール・データ・ソース

4. JDBCドライバのディレクトリ・ロケーションを設定し、[次へ]ボタンをクリックします。

セル=localhostNode01Cell, プロファイル=AppSrv01 ページを閉じる

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

ステップ 1: 新規 JDBC プロバイダーの作成

→ ステップ 2: データベース・クラスパス情報を入力

ステップ 3: 要約

データベース・クラスパス情報を入力

JDBC プロバイダーの定義に WebSphere(R) Application Server が使用する、JDBC ドライバー・クラス・ファイルのクラスパスを設定。エンタリーは、ENTER (実行) キーを使用して区切ります。パス区切り文字 (「\」や「/」など) は使用できません。値が示されている場合

クラスパス:

`${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc4.jar`
`${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar`
`${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar`

WebSphere 変数 `${DB2UNIVERSAL_JDBC_DRIVER_PATH}` として保存される "db2jcc.jar, db2jcc_license_cisuz.jar" の

ネイティブ・ライブラリー・パス

WebSphere 変数 `${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}` として保存されるディレクトリー・ロケーション

前

5. 設定内容を確認し、[終了]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラスパス情報を入力

→ ステップ 3: 要約

要約

アクションの要約:

オプション	値
有効範囲	cells:IM8213Node01Cell:nodes:IM8213Node01:servers:s
JDBC プロバイダー名	DB2 Universal JDBC Driver Provider
説明	One-phase commit DB2 JCC provider that supports JDBC Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.
クラスパス	<code>\${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar</code> <code>\${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar</code> <code>\${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_</code>
<code>\${DB2UNIVERSAL_JDBC_DRIVER_PATH}</code>	C:/IBM/SQLLIB/java
<code>\${UNIVERSAL_JDBC_DRIVER_PATH}</code>	
ネイティブ・パス	<code>\${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}</code>
<code>\${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}</code>	
実装クラス名	com.ibm.db2.jcc.DB2ConnectionPoolDataSource

前

6. [保存]をクリックします。これで、JDBCプロバイダーの設定は終わりです。

JDBC プロバイダー

メッセージ

- ローカル構成が変更されました。
- 直接マスター構成に保存できます。
- 変更を検討してから、保存または破棄してください。
- 変更を有効にするには、サーバーの再起動が必要です。

JDBC プロバイダー

このページを使用して、JDBC プロバイダーのプロパティを編集します。JDBC プロバイダー・オブジェクトは、ご使用の環境の特定のベンダー・データベースにアクセスするための特定の JDBC ドライバー実装クラスをカプセル化します。このタスクの詳細については、[ガイド付きアクティビティ](#)で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=**IM8213Node01**, サーバー=**server1**

設定

新規作成... 削除

選択 名前 有効範囲 説明

管理できるリソース:

☐	DB2 Universal JDBC Driver Provider	ノード=IM8213Node01,サーバー=server1	One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.
☐	Derby JDBC Provider	ノード=IM8213Node01,サーバー=server1	Derby embedded non-XA JDBC Provider

合計 2

Microsoft SQL Serverの設定

Microsoft SQL Serverの設定を行います。

- メニューから[リソース]-[JDBC]-[JDBC プロバイダー] を選択します。

表示: すべてのタスク

- ようこそ
- ガイド付きアクティビティ
- サーバー
- アプリケーション
- サービス
- リソース
 - スケジューラー
 - オブジェクト・プール・マネージャー
 - JMS
 - JDBC
 - JDBC プロバイダー**
 - データソース (WebSphere Application Server V4)
 - リソース・アダプター
 - 非同期 Bean
 - キャッシュ・インスタンス
 - メール
 - URL
 - リソース環境

- 「JDBC プロバイダー」画面で [新規作成] ボタンをクリックします。

JDBC プロバイダー

このページを使用して、JDBC プロバイダーのプロパティを編集します。JDBC プロバイダー・オブジェクトは、ご使用の環境の特定のベンダー・データベースにアクセスするための特定の JDBC ドライバー実装クラスをカプセル化します。このタスクの詳細については、[ガイド付きアクティビティ](#)で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=**IM8213Node01**, サーバー=**server1**

設定

新規作成... 削除

選択 名前 有効範囲 説明

管理できるリソース:

☐	Derby JDBC Provider	ノード=IM8213Node01,サーバー=server1	Derby embedded non-XA JDBC Provider
--------------------------	-------------------------------------	-------------------------------	-------------------------------------

合計 1

3. [データベース・タイプ]、[プロバイダー・タイプ]、[実装タイプ]を設定し、[次へ]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

→ ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラスパス情報を入力

ステップ 3: 要約

データベースのアクセスに必要な固有のベンダー JDBC ドライバー実装クラスをカプセル化する、JDBC プロバイダーの基本構成値

有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* データベース・タイプ
SQL Server

* プロバイダー・タイプ
Microsoft SQL Server JDBC Driver

* 実装タイプ
接続プール・データ・ソース

* 名前
Microsoft SQL Server JDBC Driver

説明
Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.

次へ キャンセル

設定項目	値
データベース・タイプ	SQL Server
プロバイダー・タイプ	Microsoft SQL Server JDBC Driver
実装タイプ	接続プール・データ・ソース

4. JDBCドライバのディレクトリ・ロケーションを設定し、[次へ]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

ステップ 1: 新規 JDBC プロバイダーの作成

→ ステップ 2: データベース・クラスパス情報を入力

ステップ 3: 要約

データベース・クラスパス情報を入力

JDBC プロバイダーの定義に WebSphere(R) Application Server が使用する、JDBC ドライバー・クラス・ファイルのクラスパスエントリは、ENTER (実行) キーを使用して区切ります。パス区切り文字 (「:」や「\」など) は使用できません。値が示されている

クラスパス:
\${MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc4.jar

適用

WebSphere 変数 \${MICROSOFT_JDBC_DRIVER_PATH} として保存される "sqljdbc4.jar" のディレクトリ・ロケーション
c:/jdbc_driver

ネイティブ・ライブラリー・パス
WebSphere 変数 \${MICROSOFT_JDBC_DRIVER_NATIVEPATH} として保存されるディレクトリ・ロケーション

前へ **次へ** キャンセル

5. 設定内容を確認し、[終了]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラスパス情報を入力

→ ステップ 3: 要約

要約

アクションの要約:

オプション	値
有効範囲	cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1
JDBC プロバイダー名	Microsoft SQL Server JDBC Driver
説明	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
クラスパス	\${MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc4.jar
ネイティブパス	\${MICROSOFT_JDBC_DRIVER_NATIVEPATH}
実装クラス名	com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource

前 | **終了** | キャンセル

6. [保存]をクリックします。これで、JDBCプロバイダーの設定は終わりです。

JDBC プロバイダー

メッセージ

- ローカル構成が変更されました。
- 直接マスター構成に**保存**します。
- 変更を**検討**してから、保存または破棄してください。

変更を有効にするには、サーバーの再起動が必要です。

JDBC プロバイダー

このページを使用して、JDBC プロバイダーのプロパティを編集します。JDBC プロバイダー・オブジェクトは、ご使用の環境の特定のベンダー・データベースにアクセスするための特定の JDBC ドライバー実装クラスをカプセル化します。このタスクの詳細については、[ガイド付きアクティビティ](#) で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=**IM8213Node01**, サーバー=**server1**

設定

新規作成... | 削除

選択 | 名前 | 有効範囲 | 説明

管理できるリソース:	名前	有効範囲	説明
☐	Derby JDBC Provider	ノード=IM8213Node01,サーバー=server1	Derby embedded non-XA JDBC Provider.
☐	Microsoft SQL Server JDBC Driver	ノード=IM8213Node01,サーバー=server1	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.

合計 2

PostgreSQLの設定

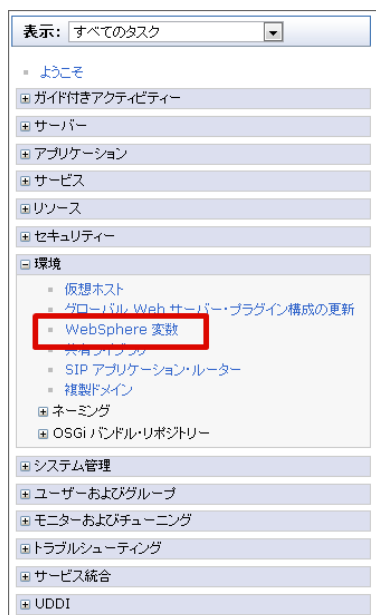
項目

- WebSphere 変数の設定
- JDBC プロバイダー

WebSphere 変数の設定

WebSphere 変数の設定を行います。

- メニューから[環境]-[WebSphere 変数]を選択します。



2. 一覧から[User-defined_JDBC_DRIVER_PATH] を選択します。

☐	SERVER_LOG_ROOT	\${LOG_ROOT}/server1	ノード =localhostNode01,サ ーバー=server1
☐	SYBASE_JDBC_DRIVER_PATH		ノード =localhostNode01
☐	UNIVERSAL_JDBC_DRIVER_PATH	\${WAS_INSTALL_ROOT}/universalDriver/lib	ノード =localhostNode01
☐	USER_INSTALL_ROOT	/opt/IBM/WebSphere/AppServer/profiles/AppSrv01	ノード =localhostNode01
☐	User-defined_JDBC_DRIVER_PATH		ノード =localhostNode01
☐	WAS_CELL_NAME	localhostNode01Cell	セル =localhostNode01Cell
☐	WAS_ETC_DIR	\${USER_INSTALL_ROOT}/etc	ノード =localhostNode01

3. 「WebSphere 変数」画面で [値] に PostgreSQL の JDBCドライバが格納されているディレクトリパスを入力し、[OK] ボタンをクリックします。

WebSphere 変数

WebSphere 変数 > User-defined_JDBC_DRIVER_PATH

このページを使用して、置換変数を定義します。この変数は、ファイルシステムのルート・ディレクトリなどの、システム定義値の間接指示のレベルを指定します。変数には、サーバー、ノード、クラスターまたはセルの有効範囲レベルがあります。1つの有効範囲レベルの値は、他のレベルの値を異なる値にできます。変数が有効範囲値と矛盾する場合は、より細かい有効範囲値が、より大きな有効範囲レベルの値をオーバーライドします。すなわちサーバー変数はノード変数をオーバーライドし、ノード変数はクラスター変数をオーバーライドし、クラスター変数はセル変数をオーバーライドします。

構成

一般プロパティ

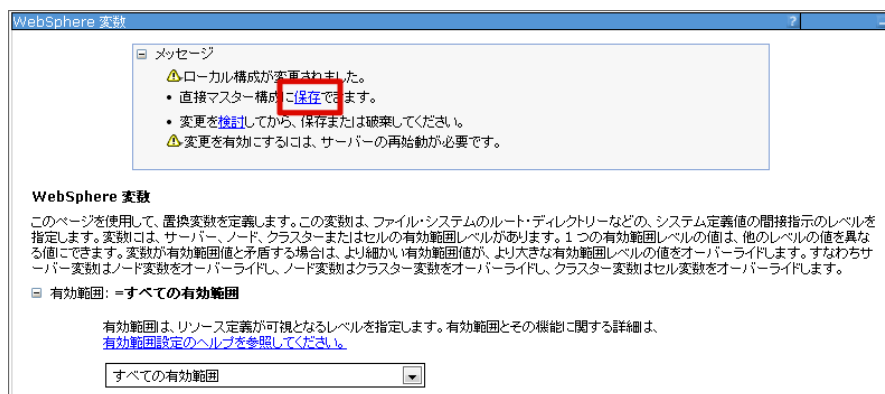
* 名前
User-defined_JDBC_DRIVER_PA

値
C:/jdbc_driver

説明
The directory that contains a JDBC 2.0 compliant driver.

適用 OK リセット キャンセル

4. [保存]をクリックします。これで、WebSphere 変数の設定は終わりです。



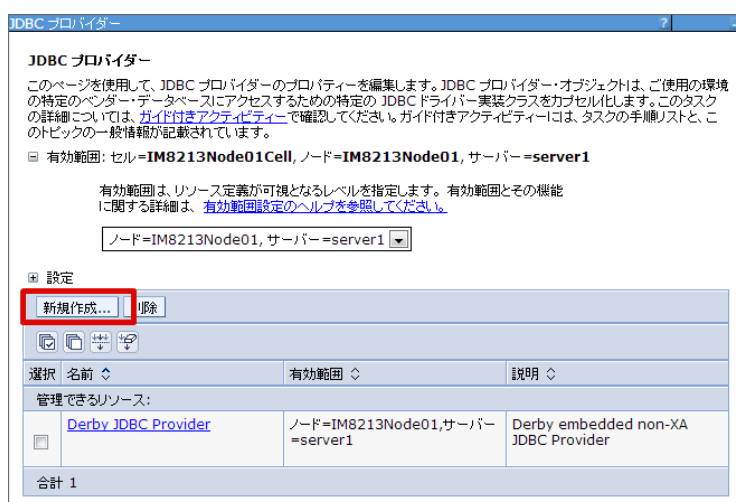
JDBC プロバイダー

JDBC プロバイダーの設定を行います。

1. メニューから[リソース]-[JDBC]-[JDBC プロバイダー]を選択します。



2. 「JDBC プロバイダー」画面で [新規作成] ボタンをクリックします。



3. [データベース・タイプ]、[プロバイダー・タイプ]、[実装タイプ]を設定し、[次へ]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

→ ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラスパス情報を入力

ステップ 3: 要約

データベースのアクセスに必要な固有のベンダー JDBC ドライバー実装クラスをカプセル化する、JDBC プロバイダーの基本構成値

有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* データベース・タイプ
ユーザー定義

* 実装クラス名
org.postgresql.ds.PGConnectionPoolData

* 名前
PostgreSQL JDBC Provider

説明
Custom JDBC2.0-compliant Provider configuration

次へ キャンセル

設定項目	値
データベース・タイプ	ユーザー定義
プロバイダー・タイプ	org.postgresql.ds.PGConnectionPoolDataSource

4. JDBCドライバのディレクトリ・ロケーションを設定し、[次へ]ボタンをクリックします。
JDBCドライバのファイル名が「 **postgresql-9.2-1002.jdbc4.jar** 」の場合は下記のように設定します。

```
${User-defined_JDBC_DRIVER_PATH}/postgresql-9.2-1002.jdbc4.jar
```

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

ステップ 1: 新規 JDBC プロバイダーの作成

→ ステップ 2: データベース・クラスパス情報を入力

ステップ 3: 要約

データベース・クラスパス情報を入力

ユーザー定義の JDBC プロバイダーを構成するには、インストールした JDBC ドライバー・クラス・ファイルの絶対パス名を指定します。

クラスパス
\${User-defined_JDBC_DRIVER_PATH}/postgresql-9.2-1002.jdbc4.jar

前へ 次へ キャンセル

5. 設定内容を確認し、[終了]ボタンをクリックします。

新規 JDBC プロバイダーの作成

新規 JDBC プロバイダーの作成

ステップ 1: 新規 JDBC プロバイダーの作成

ステップ 2: データベース・クラスパス情報を入力

→ ステップ 3: 要約

要約

アクションの要約:

オプション	値
有効範囲	cells:localhostNode01Cell:nodes:localhostNode01:servers:server1
JDBC プロバイダー名	PostgreSQL JDBC Provider
説明	Custom JDBC2.0-compliant Provider configuration
クラスパス	\${User-defined_JDBC_DRIVER_PATH}/postgresql-9.2-1002.jdbc4.jar
実装クラス名	org.postgresql.ds.PGConnectionPoolDataSource

前へ 終了 キャンセル

6. [保存]をクリックします。これで、JDBCプロバイダーの設定は終わりです。

JDBC プロバイダー

メッセージ

- ローカル構成が変更されました。
 - 直接マスター構成に保存されます。
 - 変更を[検証](#)してから、保存または破棄してください。
- 変更を有効にするには、サーバーの再起動が必要です。

JDBC プロバイダー

このページを使用して、JDBC プロバイダーのプロパティを編集します。JDBC プロバイダー・オブジェクトは、ご使用の環境の特定のベンダー・データベースにアクセスするための特定の JDBC ドライバー実装クラスをカプセル化します。このタスクの詳細については、[ガイド付きアクティビティ](#)で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=IM8213Node01, サーバー=server1

設定

新規作成... 削除

選択 名前 有効範囲 説明

管理できるリソース:

☐	Derby JDBC Provider	ノード=IM8213Node01,サーバー=server1	Derby embedded non-XA JDBC Provider.
☐	PostgreSQL JDBC Provider	ノード=IM8213Node01,サーバー=server1	Custom JDBC2.0-compliant Provider configuration

合計 2

データソースの設定

項目

- データ・ソースの登録
- データ・ソースのカスタム・プロパティの設定

データ・ソースの登録

データソースの登録を行います。

- メニューから[リソース]-[JDBC]-[データ・ソース] を選択します。

表示: すべてのタスク

- ようこそ
- ガイド付きアクティビティ
- サーバー
- アプリケーション
- サービス
- リソース
 - スケジューラー
 - オブジェクト・プール・マネージャー
 - JMS
 - JDBC
 - JDBC プロバイダー
 - データ・ソース**
 - データベース (WebSphere Application Server V4)
 - リソース・アダプター
 - 非同期 Bean
 - キャッシュ・インスタンス
 - メール
 - URL
 - リソース環境

- 「データ・ソース」画面で [新規作成] ボタンをクリックします。

データ・ソース

データ・ソース

このページを使用して、選択済み JDBC プロバイダーに関連付けられたデータ・ソースの設定を編集します。データ・ソース・オブジェクトは、データベースにアクセスするための接続をアプリケーションに提供します。このタスクの詳細については、[ガイド付きアクティビティ](#)で確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプ](#)を参照してください。

ノード=**IM8213Node01**, サーバー=**server1**

設定

新規作成... 削除 テスト接続 状態の管理...

選択 名前 JNDI 名 有効範囲 プロバイダー 説明 カテゴリ

なし。

合計 0

3. [データ・ソース名]と[JNDI名]を入力し、[次へ] ボタンをクリックします。
ここでは [データ・ソース名] を「intra-mart DEFAULT」、[JNDI名] を「jdbc/default」とします。

データ・ソースの作成

データ・ソースの作成

→ ステップ 1: 基本データ・ソース情報の入力

基本データ・ソース情報の入力

JDBC プロバイダーとのアソシエーションに対するデータ・ソースの基本構成値を設定します。データ・ソースは、アプリケーション・要件: アプリケーションが Enterprise JavaBeans(TM) (EJB) 1.0 仕様または Java(TM) Servlet 2.2 仕様に基づく場合、有効範囲

cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

データベース名

intra-mart DEFAULT

JNDI 名

jdbc/default

ステップ 5: 要約

次へ キャンセル



コラム

[JNDI名]は intra-mart Accel Platform 構成ファイルの [DataSourceマッピングの設定](#) の **data-source-mapping-config.xml** において、**<resource-ref-name>** に記述したJNDI名を入力してください。

4. [JDBCプロバイダの設定](#)で作成したJDBCプロバイダーを選択し、[次へ] ボタンをクリックします。

データ・ソースの作成

データ・ソースの作成

→ ステップ 2: JDBCプロバイダーを選択

JDBC プロバイダーを選択

データ・ソースをサポートする JDBC プロバイダーを指定します。新規 JDBC プロバイダーの作成を選択すると、同じ有効範囲

☐ 新規 JDBC プロバイダーの作成

☒ 既存 JDBC プロバイダーを選択

選択...

選択...

Derby JDBC Provider

PostgreSQL JDBC Provider

Oracle JDBC Driver

DB2 Universal JDBC Driver Provider

Microsoft SQL Server JDBC Driver

次へ キャンセル

5. データ・ソースのデータベース特定プロパティを入力し、[次へ] ボタンをクリックします。
- Oracleの場合
[URL]、[データ・ストアのヘルパー・クラス名]を入力します。
[データ・ストアのヘルパー・クラス名]は「Oracle11g データ・ストア・ヘルパー」を選択してください。

データソースの作成

データソースの作成

ステップ 1: 基本データソース情報の入力

ステップ 2: JDBC プロバイダーを選択

→ ステップ 3: データソースのデータベース特定プロパティを入力

ステップ 4: セキュリティー別名のセットアップ

ステップ 5: 要約

データソースのデータベース特定プロパティを入力

データソースによって管理される接続をサポートするために、データベースベンダーの JDBC ドライバーが必要とする、これらの

名前	値
* URL	jdbc:oracle:thin:@localhost:152
* データストアのヘルパー・クラス名	Oracle11g データストアヘルパー

☒ コンテナ管理バースタンス (CMP) 内でこのデータソースを使用する

前へ 次へ キャンセル

- DB2の場合

[ドライバー・タイプ]、[データベース名]、[サーバー名]、[ポート番号]を入力します。

[ドライバー・タイプ] は「4」を選択してください。

データソースの作成

データソースの作成

ステップ 1: 基本データソース情報の入力

ステップ 2: JDBC プロバイダーを選択

→ ステップ 3: データソースのデータベース特定プロパティを入力

ステップ 4: セキュリティー別名のセットアップ

ステップ 5: 要約

データソースのデータベース特定プロパティを入力

データソースによって管理される接続をサポートするために、データベースベンダーの JDBC ドライバーが必要とする、これらの

名前	値
* ドライバー・タイプ	4
* データベース名	dbname
* サーバー名	localhost
* ポート番号	50000

☒ コンテナ管理バースタンス (CMP) 内でこのデータソースを使用する

前へ 次へ キャンセル

- SQL Serverの場合

[データベース名]、[ポート番号]、[サーバー名]を入力します。

データソースの作成

データソースの作成

ステップ 1: 基本データソース情報の入力

ステップ 2: JDBC プロバイダーを選択

→ ステップ 3: データソースのデータベース特定プロパティを入力

ステップ 4: セキュリティー別名のセットアップ

ステップ 5: 要約

データソースのデータベース特定プロパティを入力

データソースによって管理される接続をサポートするために、データベースベンダーの JDBC ドライバーが必要とする、これらの

名前	値
データベース名	dbname
ポート番号	1433
サーバー名	localhost

☒ コンテナ管理バースタンス (CMP) 内でこのデータソースを使用する

前へ 次へ キャンセル

- PostgreSQLの場合

[データストアのヘルパー・クラス名]に、「com.ibm.websphere.rsadapter.GenericDataStoreHelper」を入力してください。

データソースの作成

データソースの作成

ステップ 1: 基本データソース情報の入力

ステップ 2: JDBC プロバイダーを選択

→ ステップ 3: データソースのデータベース特定プロパティを入力

ステップ 4: セキュリティー別名のセットアップ

ステップ 5: 要約

データソースのデータベース特定プロパティを入力

ユーザー定義のデータソースに対して、データベースベンダーの JDBC ドライバーに必要なプロパティを指定します。必要な

名前	値
* データストアのヘルパー・クラス名	com.ibm.websphere.rsadapter.GenericDataStoreHelper

☒ コンテナ管理バースタンス (CMP) 内でこのデータソースを使用する

前へ 次へ キャンセル



コラム

GenericDataStoreHelperを利用しているため、以下の警告がDB利用時に出力されます。

「警告: GenericDataStoreHelper が使用されています。」

データソースの作成

データソースの作成

ステップ 1: 基本データソース情報の入力

ステップ 2: JDBC プロバイダーを選択

ステップ 3: データソースのデータベース特定プロパティを入力

→ ステップ 4: セキュリティー別名のセットアップ

ステップ 5: 要約

セキュリティ別名のセットアップ

このリソースに認証される値の選択

コンポーネント管理認証別名
IM8213Node01/db_user

マッピング構成別名
(なし)

コンテナ管理認証別名
(なし)

注: 新しい J2C 認証別名は、次のリンクのいずれかにアクセスすることによって作成できます。リンクをクリックすると、ウィザード

[グローバル J2C 認証別名](#)
[セキュリティドメイン](#)

前 次へ キャンセル

7. 設定内容を確認し、[終了]ボタンをクリックします。

データソースの作成

データソースの作成

ステップ 1: 基本データソース情報の入力

ステップ 2: JDBC プロバイダーを選択

ステップ 3: データソースのデータベース特定プロパティを入力

ステップ 4: セキュリティー別名のセットアップ

→ ステップ 5: 要約

要約

アクションの要約:

オプション	値
有効範囲	cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1
データソース名	intra-mart DEFAULT
JNDI 名	jdbc/default
既存 JDBC プロバイダーを選択	Oracle JDBC Driver
実装クラス名	oracle.jdbc.pool.OracleConnectionPoolDataSource
URL	jdbc:oracle:thin:@localhost:1521:dbname
データストアのヘルパークラス名	com.ibm.websphere.rsadapter.Oracle11gDataStoreHelper
コンテナ管理バースタンス (CMP) 内でこのデータソースを使用する	true
コンポーネント管理認証別名	IM8213Node01/db_user
マッピング構成別名	(なし)
コンテナ管理認証別名	(なし)

前 終了 キャンセル

8. [保存]をクリックします。Oracle、DB2を使用する場合はこれで終了です。
SQL Server、PostgreSQL を使用する場合は、次の [データソースのカスタム・プロパティの設定](#) も行ってください。

データ・ソース

メッセージ

- ローカル構成が変更されました。
- 直接マスター構成に保存できます。
- 変更を検討してから、保存または破棄してください。
- 変更を有効にするには、サーバーの再始動が必要です。

データ・ソース

このページを使用して、選択済み JDBC プロバイダーに関連付けられたデータ・ソースの設定を編集します。データ・ソース・オブジェクトは、データベースにアクセスするための接続をアプリケーションに提供します。このタスクの詳細については、[ガイド付きアクティビティ](#)を確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=**IM8213Node01**, サーバー=**server1**

設定

新規作成... 削除 テスト接続 状態の管理...

選択	名前	JNDI 名	有効範囲	プロバイダー	説明	カテゴリ
☐	intra-mart DEFAULT	jdbc/default	ノード= IM8213Node01 , サーバー= server1	Oracle JDBC Driver	New JDBC Datasource	

合計 1

データ・ソースのカスタム・プロパティの設定

カスタム・プロパティの設定を行います。

カスタム・プロパティの設定は **SQL Server**、**PostgreSQL** を使用する場合に必要です。

- メニューから[リソース]-[JDBC]-[データ・ソース]を選択し、先ほど登録したデータ・ソース名「intra-mart DEFAULT」のリンクをクリックします。

データ・ソース

このページを使用して、選択済み JDBC プロバイダーに関連付けられたデータ・ソースの設定を編集します。データ・ソース・オブジェクトは、データベースにアクセスするための接続をアプリケーションに提供します。このタスクの詳細については、[ガイド付きアクティビティ](#)を確認してください。ガイド付きアクティビティには、タスクの手順リストと、このトピックの一般情報が記載されています。

有効範囲: セル=**IM8213Node01Cell**, ノード=**IM8213Node01**, サーバー=**server1**

有効範囲は、リソース定義が可視となるレベルを指定します。有効範囲とその機能に関する詳細は、[有効範囲設定のヘルプを参照してください](#)。

ノード=**IM8213Node01**, サーバー=**server1**

設定

新規作成... 削除 テスト接続 状態の管理...

選択	名前	JNDI 名	有効範囲	プロバイダー	説明	カテゴリ
☐	intra-mart DEFAULT	jdbc/default	ノード= IM8213Node01 , サーバー= server1	Microsoft SQL Server JDBC Driver	Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable in version 6.1.0.15 and later nodes.	

合計 1

- [追加プロパティ]-[カスタム・プロパティ]を選択します。

データソース

データソース > intra-mart DEFAULT

このページを使用して、選択済み JDBC プロバイダーに関連付けられたデータソースの設定を編集します。データソース・オブジェクトは、データベースにアクセスするための接続をアプリケーションに提供します。

構成

テスト接続

一般プロパティ

* 有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* プロバイダー
Microsoft SQL Server JDBC Driver

* 名前
intra-mart DEFAULT

JNDI 名
jdbc/default

☒ コンテナ管理バースタンス (CMP) 内でこのデータソースを使用する

説明
Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable in version 6.1.0.15 and later nodes.

追加プロパティ

接続プール・プロパティ

WebSphere Application Server データソース・プロパティ

カスタム・プロパティ

関連項目

JAAS - J2C 認証データ

3. カスタム・プロパティを設定します。使用するデータベースに応じて編集・追加してください

■ SQL Serverの場合

[selectMethod] リンクをクリックします。

			driver documentation for more information.	
☐	selectMethod	direct	Specifies the type of cursors uses for SQL query results. Values are 'direct' (the default) or 'cursor'. Refer to the JDBC driver documentation for more information.	false
☐	sendStringParametersAsUnicode	false	Specifies how string parameters are transmitted to the database, as unicode or in native format for the database.	false

値に「Cursor」を設定し、[OK]をクリックします。

データソース

データソース > intra-mart DEFAULT > カスタム・プロパティ > selectMethod

このページを使用して、構成するリソース・プロバイダーおよびリソース・ファクトリーに対してエンタープライズ情報システム (EIS) が必要とするカスタム・プロパティは、データベースにアクセスするデータソースの追加カスタム・プロパティを必要とします。

構成

一般プロパティ

* 有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

☐ 必要

名前
selectMethod

値
Cursor

説明
Specifies the type of cursors uses for SQL query results. Values are 'direct' (the default) or 'cursor'. Refer to the JDBC driver

タイプ
java.lang.String

適用 OK リセット キャンセル

■ PostgreSQLの場合

下記のプロパティを追加します。

- serverName
- databaseName
- portNumber

[新規作成] ボタンをクリックします。

データ・ソース

データ・ソース > intra-mart DEFAULT > カスタム・プロパティ

このページを使用して、構成するリソース・プロバイダーおよびリソース・ファクトリーに対してエンタープライズ情報システム (EIS) が必要とするカスタム・プロパティを追加します。

設定

新規作成... 削除

選択 名前 値 説明

管理できるリソース:

☐	freeResourcesOnClose	false	Controls whether or not the application server automatically free Readers when the object that created them is closed. The ability the free (or close) method.
☐	userDefinedErrorMap		Overlays existing entries in the error map by invoking DataStore used to add, change, or remove entries from the error map. Entry and value, where the key is an error code (numeric value) or SQLState separated by = (equals sign). For example, to remove the mapping DuplicateKeyException, and add a mapping of SQLState 08004 to userDefinedErrorMap: "S1000"=;1062=com.ibm.websphere.ce.cm.DuplicateKeyException

プロパティを追加してください。

データ・ソース

データ・ソース > intra-mart DEFAULT > カスタム・プロパティ > 新規作成...

このページを使用して、構成するリソース・プロバイダーおよびリソース・ファクトリーに対してエンタープライズ情報システム (EIS) が必要とするカスタム・プロパティを指定します。例えば、多くのデータベース・ベンダーは、データベースにアクセスするデータ・ソースの追加カスタム・プロパティを必要とします。

構成

一般プロパティ

* 有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* 名前
serverName

値
localhost

説明

タイプ
java.lang.String

適用 OK セット キャンセル

データ・ソース

データ・ソース > intra-mart DEFAULT > カスタム・プロパティ > 新規作成...

このページを使用して、構成するリソース・プロバイダーおよびリソース・ファクトリーに対してエンタープライズ情報システム (EIS) が必要とするカスタム・プロパティを指定します。例えば、多くのデータベース・ベンダーは、データベースにアクセスするデータ・ソースの追加カスタム・プロパティを必要とします。

構成

一般プロパティ

* 有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* 名前
databaseName

値
dbname

説明

タイプ
java.lang.String

適用 OK セット キャンセル

データソース

データソース > intra-mart DEFAULT > カスタム・プロパティ > 新規作成...

このページを使用して、構成するリソース・プロバイダーおよびリソース・ファクトリーに対してエンタープライズ情報システム (EIS) が必要とするカスタム・プロパティを指定します。例えば、多くのデータベース・ベンダーは、データベースにアクセスするデータソースの追加カスタム・プロパティを必要とします。

構成

一般プロパティ

* 有効範囲
cells:IM8213Node01Cell:nodes:IM8213Node01:servers:server1

* 名前
portNumber

値
5432

説明

タイプ
java.lang.String

適用 OK リセット キャンセル

4. [保存]をクリックします。これで、カスタム・プロパティの設定は終わりです。

データソース

メッセージ

- ローカル構成が変更されました。
- 直接マスター構成に保存できます。
- 変更を検討してから、保存または破棄してください。
- 変更を有効にするには、サーバーの再始動が必要です。

データソース > intra-mart DEFAULT > カスタム・プロパティ

このページを使用して、構成するリソース・プロバイダーおよびリソース・ファクトリーに対してエンタープライズ情報システム (EIS) が必要とするカスタム・プロパティを指定します。例えば、多くのデータベース・ベンダーは、データベースにアクセスするデータソースの追加カスタム・プロパティを必要とします。

設定

新規作成... 削除

選択 名前 値 説明

管理できるリソース:	名前	値	説明
☐	freeResourcesOnClose	false	Controls whether or not the application server automatically free Readers when the object that created them is closed. The ability the free (or close) method.

warファイルのデプロイ

warファイルのデプロイを行います。

1. メニューから[アプリケーション]-[新規アプリケーション]を選択し、[新規エンタープライズ・アプリケーション]リンクをクリックします。

WebSphere, software

ようこそ ヘルプ ログアウト IBM

セル=IM8213Node01Cell, プロファイル=AppSrv01

ページを閉じる

表示: すべてのタスク

- ようこそ
- ガイド付きアクティビティ
- サーバー
- アプリケーション
 - 新規アプリケーション**
 - アプリケーション・タイプ
 - グローバル・デプロイメント設定
- サービス
- リソース
- セキュリティ
- 環境
- システム管理
- ユーザーおよびグループ
- モニターおよびチューニング
- トラブルシューティング
- サービス統合
- UDDI

新規アプリケーション

新規アプリケーション

このページから、異なるタイプの新規アプリケーションを作成することができます。

新規アプリケーションのインストール

- 新規エンタープライズ・アプリケーション
- 新規ビジネス・レベル・アプリケーション
- 新規アセット

2. [新規アプリケーションへのパス]を設定します。

[リモート・ファイル・システム]の[絶対パス]にwarファイルのフルパスを入力し、[次へ]ボタンをクリックします。

アプリケーション・インストールの準備

アップロードおよびインストールする EAR、WAR、JAR、または SAR モジュールを指定します。

新規アプリケーションへのパス

☐ ローカル・ファイル・システム
絶対パス
ファイルを選択 選択されていません

☒ リモート・ファイル・システム
絶対パス
C:\imart.war 参照 ...

次へ キャンセル

3. [次へ]ボタンをクリックします。

アプリケーション・インストールの準備

アプリケーションをどのようにインストールしますか？

☒ ファスト・パス - 追加情報が必要な場合のみプロンプトを出す

☐ 詳細 - すべてのインストール・オプションおよびパラメーターを表示する

☒ デフォルトのマッピングおよびマッピングの生成を選択します。

前 次へ キャンセル

4. [次へ]ボタンをクリックします。

新規アプリケーションのインストール

エンタープライズ・アプリケーションおよびモジュールのインストールのオプションを指定してください。

ステップ 1: インストール・オプションの選択

インストール・オプションの選択

アプリケーションに使用可能なさまざまなオプションを指定します。

☐ JavaServer Pages ファイルのプリコンパイル

アプリケーションをインストールするディレクトリ

☒ アプリケーションを配付する

☐ バイナリー構成の使用

☐ エンタープライズ Bean のデプロイ

アプリケーション名

imart_war

☒ リソース用の MBean を作成する

☐ Web および EJB モジュールのクラス再ロード設定をオーバーライドする

再ロード間隔 (秒)

☐ Web サービスをデプロイする

入力のオフ/警告/失敗の検証

警告

☐ プロセス組み込み構成

ファイル・アクセス権

すべてのファイルの読み取りを許可するが、書き込みは許可しない
実行可能ファイルの実行を許可する
HTML およびイメージ・ファイルの読み取りを全員に許可する

.*\dll=755#.*\so=755#.*\a=755#.*\sl=755

アプリケーション・ビルド ID

不明

☐ リモート・リソースに対する組み込みディスパッチを許可する

☐ リモート・リソースからの組み込みサービスを許可する

ビジネス・レベル・アプリケーション名

新規 BLA の作成

非同期要求ディスパッチ・タイプ

使用不可

☐ EJB 参照ターゲットを自動で解決できるようにする

☐ クライアント・モジュールをデプロイする

クライアント・デプロイメント・モード

分離

☐ スキーマを検証する

次へ キャンセル

5. [次へ]ボタンをクリックします。

新規アプリケーションのインストール

エンタープライズ・アプリケーションおよびモジュールのインストールのオプションを指定してください。

ステップ 1 インストール・オプションの選択

→ ステップ 2: モジュールをサーバーにマップ

★ ステップ 3 リソース参照をリソースにマップ

★ ステップ 4 Web モジュール用の仮想ホストをマップ

★ ステップ 5 Web モジュールのコンテキスト・ルートマップ

★ ステップ 6 モジュールのメタデータ

ステップ 7 要約

モジュールをサーバーにマップ

アプリケーションに含まれるモジュールをインストールする。アプリケーション・サーバーやアプリケーション・サーバーのクラスターなどのターゲットを指定します。モジュールは同じアプリケーション・サーバー上にインストールすることも、複数のアプリケーション・サーバーに分散させることもできます。また、このアプリケーションに対する要求のルーターとして機能するターゲットとして、Web サーバーを指定します。各 Web サーバーのプラグイン構成ファイル (plugin-cfg.xml) は、経由して経路指定されるアプリケーションに基づいて生成されます。

クラスターおよびサーバー:

WebSphere:cell=IM8213Node01Cell,node=IM8213Node01,server=server1 適用

選択	モジュール	URI	サーバー
☐	imart.war	imart.war,WEB-INF/web.xml	WebSphere:cell=IM8213Node01Cell,node=IM8213Node01,server=server1

前へ 次へ キャンセル

6. ワークマネージャのリソース参照をWebSphereのリソースにマップします。
ここでは下記のようにマップします。

リソースタイプ	リソース参照	ターゲット・リソース JNDI 名
commonj.work.WorkManager	wm/WorkManager	wm/default

新規アプリケーションのインストール

エンタープライズ・アプリケーションおよびモジュールのインストールのオプションを指定してください。

ステップ 1 インストール・オプションの選択

ステップ 2 モジュールをサーバーにマップ

→ ステップ 3: リソース参照をリソースにマップ

★ ステップ 4 Web モジュール用の仮想ホストをマップ

★ ステップ 5 Web モジュールのコンテキスト・ルートマップ

★ ステップ 6 モジュールのメタデータ

ステップ 7 要約

リソース参照をリソースにマップ

アプリケーションに定義された各リソース参照をリソースにマップする必要があります。

commonj.work.WorkManager

複数の JNDI 名を設定

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名
☐	imart.war		imart.war,WEB-INF/web.xml	wm/WorkManager	wm/default 参照 ...

javax.sql.DataSource

複数の JNDI 名を設定 リソース認証方式の変更... 拡張プロパティ...

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名	ログイン構成
☐	imart.war		imart.war,WEB-INF/web.xml	jdbc/default	参照 ...	リソース許可: コンテナー 認証方式: なし。

前へ 次へ キャンセル

7. データソースのリソース参照をWebSphereのリソースにマップします。
ここでは下記のようにマップします。

リソースタイプ	リソース参照	ターゲット・リソース JNDI 名
javax.sql.DataSource	jdbc/default	jdbc/default

新規アプリケーションのインストール

エンタープライズ・アプリケーションおよびモジュールのインストールのオプションを指定してください。

ステップ 1 インストール・オプションの選択
ステップ 2 モジュールをサーバーにマップ
→ ステップ 3: リソース参照をリソースにマップ
★ ステップ 4 Web モジュール用の仮想ホストをマップ
★ ステップ 5 Web モジュールのコンテキスト・ルートをマップ
★ ステップ 6 モジュールのメタデータ
ステップ 7 要約

リソース参照をリソースにマップ

アプリケーションに定義された各リソース参照をリソースにマップする必要があります。

commonj.work.WorkManager

複数の JNDI 名を設定

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名
☐	imart.war		imart.war,WEB-INF/web.xml	wm/WorkManager	wm/default 参照...

javax.sql.DataSource

複数の JNDI 名を設定 | リソース認証方式の変更... | 拡張プロパティ...

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名	ログイン構成
☐	imart.war		imart.war,WEB-INF/web.xml	jdbc/default	jdbc/default 参照...	リソース許可: コンテナー 認証方式: なし。

前へ 次へ キャンセル

8. データソースのリソース認証方式を設定します。
[リソース認証方式の変更...]ボタンをクリックします。

新規アプリケーションのインストール

エンタープライズ・アプリケーションおよびモジュールのインストールのオプションを指定してください。

ステップ 1 インストール・オプションの選択
ステップ 2 モジュールをサーバーにマップ
→ ステップ 3: リソース参照をリソースにマップ
★ ステップ 4 Web モジュール用の仮想ホストをマップ
★ ステップ 5 Web モジュールのコンテキスト・ルートをマップ
★ ステップ 6 モジュールのメタデータ
ステップ 7 要約

リソース参照をリソースにマップ

アプリケーションに定義された各リソース参照をリソースにマップする必要があります。

commonj.work.WorkManager

複数の JNDI 名を設定

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名
☐	imart.war		imart.war,WEB-INF/web.xml	wm/WorkManager	wm/default 参照...

javax.sql.DataSource

複数の JNDI 名を設定 | **リソース認証方式の変更...** | 拡張プロパティ...

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名	ログイン構成
☐	imart.war		imart.war,WEB-INF/web.xml	jdbc/default	jdbc/default 参照...	リソース許可: コンテナー 認証方式: なし。

前へ 次へ キャンセル

9. 認証方式を設定します。**DBユーザ情報の設定**で登録したユーザ情報を設定してください。
ここでは下記のように選択します。
選択後、[imart.war]のチェックボックスをクリックし、[適用]ボタンをクリックします。

認証方式	認証データ入力
デフォルト・メソッドの使用 (多対1のマッピング) [ノード名]/db_user	

新規アプリケーションのインストール

エンタープライズ・アプリケーションおよびモジュールのインストールのオプションを指定してください。

ステップ 1 インストール・オプションの選択
ステップ 2 モジュールをサーバーにマップ
→ ステップ 3: リソース参照をリソースにマップ
ステップ 4 Web モジュール用の仮想ホストをマップ
ステップ 5 Web モジュールのコンテキスト・ルートをマップ
ステップ 6 モジュールのメタデータ
ステップ 7 要約

リソース参照をリソースにマップ

アプリケーションに定義された各リソース参照をリソースにマップする必要があります。

commonj.work.WorkManager

複数の JNDI 名を設定

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名
☐	imart.war		imart.war,WEB-INF/web.xml	wm/WorkManager	wm/default 参照 ...

javax.sql.DataSource

複数の JNDI 名を設定 リソース認証方式の変更... 拡張プロパティ...

認証方式の指定:

- ☐ なし。
- ☒ デフォルト・メソッドの使用 (多対 1 のマッピング)
認証データ入力: IM8213Node01/db_user
- ☐ トラストド接続の使用 (1 対 1 のマッピング)
認証データ入力: 選択...
- ☐ カスタム・ログイン構成を使用
アプリケーション・ログイン構成: 選択...

適用

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名	ログイン構成
☒	imart.war		imart.war,WEB-INF/web.xml	jdbc/default	jdbc/default 参照 ...	リソース許可: コンテナー 認証方式: なし。

前へ 次へ キャンセル

10. [次へ]ボタンをクリックします。

新規アプリケーションのインストール

エンタープライズ・アプリケーションおよびモジュールのインストールのオプションを指定してください。

ステップ 1 インストール・オプションの選択
ステップ 2 モジュールをサーバーにマップ
→ ステップ 3: リソース参照をリソースにマップ
ステップ 4 Web モジュール用の仮想ホストをマップ
ステップ 5 Web モジュールのコンテキスト・ルートをマップ
ステップ 6 モジュールのメタデータ
ステップ 7 要約

リソース参照をリソースにマップ

アプリケーションに定義された各リソース参照をリソースにマップする必要があります。

commonj.work.WorkManager

複数の JNDI 名を設定

選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名
☐	imart.war		imart.war,WEB-INF/web.xml	wm/WorkManager	wm/default 参照 ...

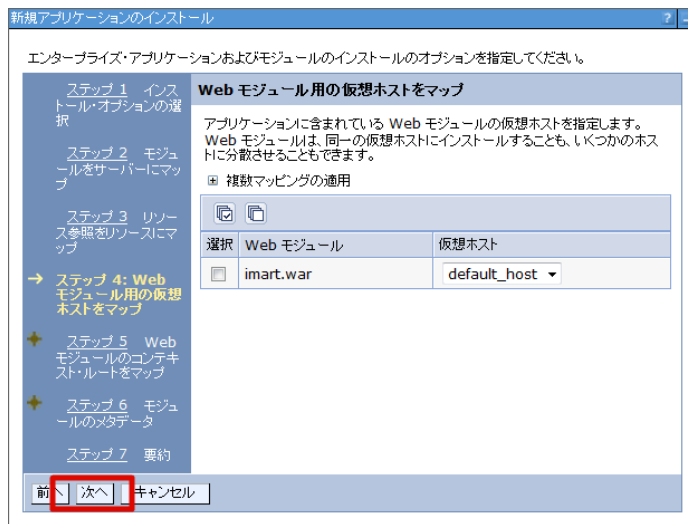
javax.sql.DataSource

複数の JNDI 名を設定 リソース認証方式の変更... 拡張プロパティ...

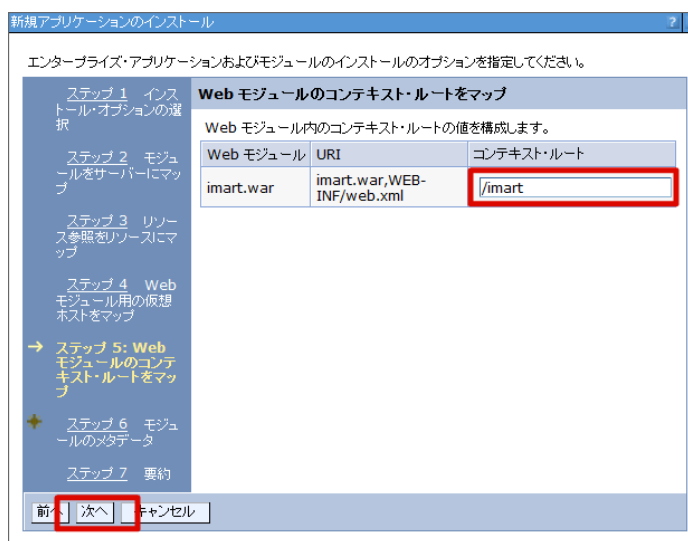
選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名	ログイン構成
☐	imart.war		imart.war,WEB-INF/web.xml	jdbc/default	jdbc/default 参照 ...	リソース許可: コンテナー 認証方式: DefaultPrincipalMapping 認証データ入力: IM8213Node01/db_user

前へ 次へ キャンセル

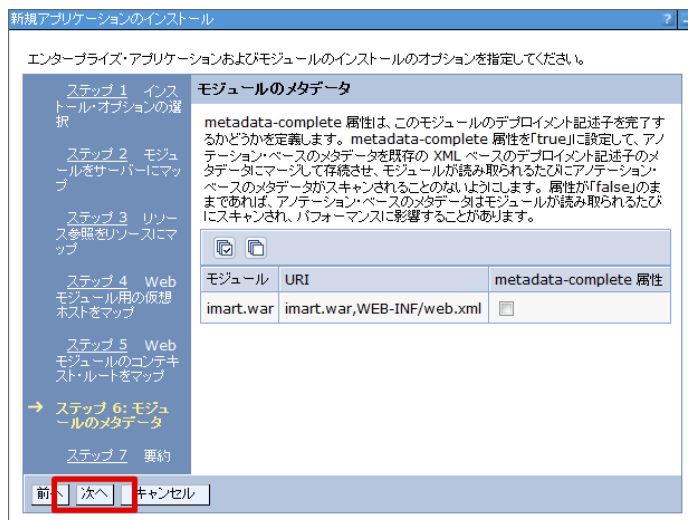
11. [次へ]ボタンをクリックします。



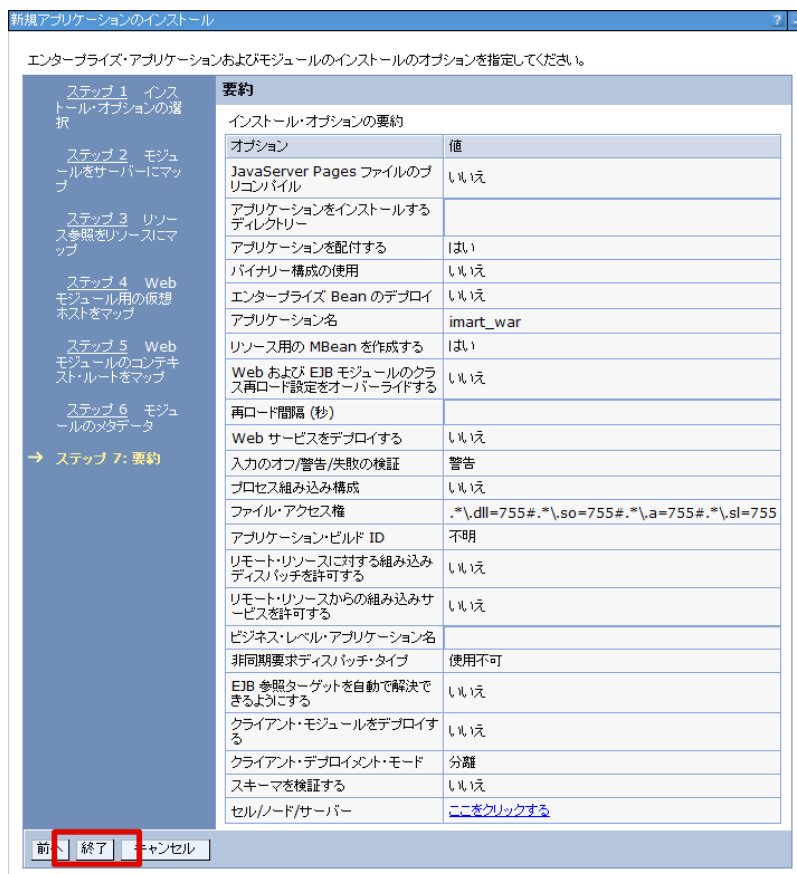
12. [コンテキスト・ルート] にコンテキストパスを入力し、[次へ]ボタンをクリックします。
ここでは「/imart」とします。



13. [次へ]ボタンをクリックします。



14. 設定内容を確認後、[終了]ボタンをクリックします。



15. 「アプリケーション imart_war は正常にインストールされました。」が表示されたら war ファイルは正常に終了です。

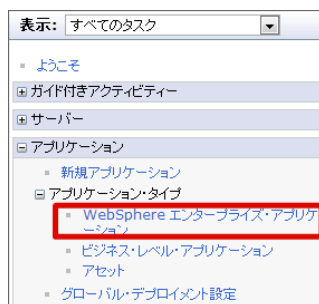
[保存]をクリックします。

続いて [クラスローダーの設定](#) を行います。

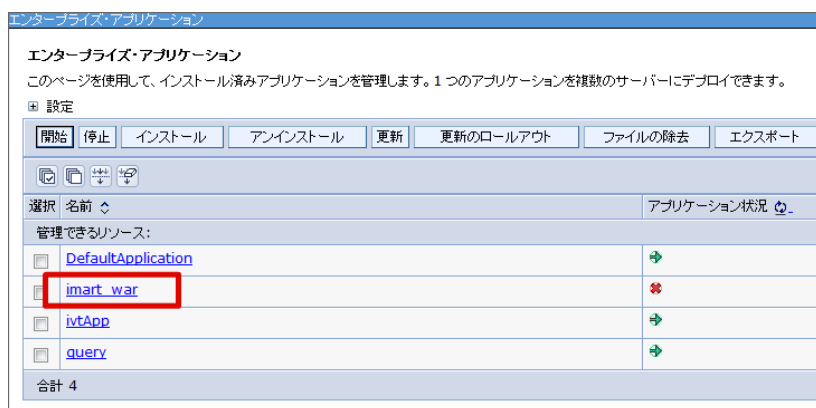


クラスローダーの設定を行います。

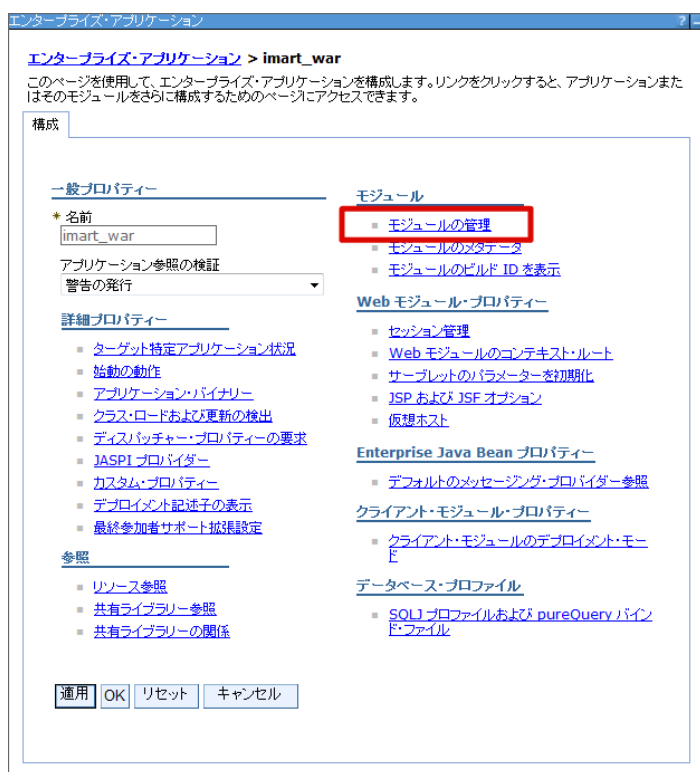
1. メニューから[アプリケーション]-[アプリケーション・タイプ]-[WebSphere エンタープライズ・アプリケーション]を選択します。



2. [imart_war] リンクをクリックします。



3. [モジュールの管理] リンクをクリックします。



4. [imart_war] リンクをクリックします。

エンタープライズ・アプリケーション

エンタープライズ・アプリケーション > imart_war > モジュールの管理

モジュールの管理

アプリケーションに含まれるモジュールをインストールする。アプリケーション・サーバーやアプリケーション・サーバーのクラスターなどのターゲットを指定します。モジュールは同じアプリケーション・サーバー上にインストールすることも、複数のアプリケーション・サーバーに分散させることもできます。また、このアプリケーションに対する要求のルーターとして機能するターゲットとして、Web サーバーを指定します。各 Web サーバーのプラグイン構成ファイル (plugin-cfg.xml) は、経由して経路指定されるアプリケーションに基づいて生成されます。

クラスターおよびサーバー:

WebSphere:cell=IM8213Node01Cell,node=IM8213Node01,server=server1 適用

除去 更新 ファイルの除去 ファイルのエクスポート

選択	モジュール	URI	モジュールタイプ	サーバー
☒	imart_war	imart_war.WEB-INF/web.xml	Web モジュール	WebSphere:cell=IM8213Node01Cell,node=IM8213Node01,server=server1

OK キャンセル

5. クラスローダーの順序を「最初にローカル・クラスローダーをロードしたクラス (親は最後)」に変更し[OK]をクリックします。

エンタープライズ・アプリケーション

エンタープライズ・アプリケーション > imart_war > モジュールの管理 > imart_war

このページを使用して、アプリケーションにデプロイ済み Web モジュールのインスタンスを構成します。このページには、セッション管理設定を含む Web モジュールのデプロイメント特定の情報が含まれています。

構成

一般プロパティ

* URI
imart_war

代替デプロイメント記述子

* 開始ウェイト
10000

* クラスローダー順序
最初にローカル・クラスローダーをロードしたクラス (親は最後) ▼

追加プロパティ

- モジュール・クラスローダーの表示
- カスタム・プロパティ
- ターゲット特定アプリケーション状況
- デプロイメント記述子の表示
- ポートレット・デプロイメント記述子の表示
- セッション管理

適用 OK リセット キャンセル

6. [保存]をクリックします。これで、クラスローダーの設定は終わりです。

エンタープライズ・アプリケーション

メッセージ

- ローカル構成が変更されました。
 - 直接マスター構成に保存できます。
 - 変更を検証してから、またはまだは破壊してください。
 - 変更を有効にするには、サーバーの再起動が必要です。

エンタープライズ・アプリケーション > imart_war > モジュールの管理

モジュールの管理

アプリケーションに含まれるモジュールをインストールする。アプリケーション・サーバーやアプリケーション・サーバーのクラスターなどのターゲットを指定します。モジュールは同じアプリケーション・サーバー上にインストールすることも、複数のアプリケーション・サーバーに分散させることもできます。また、このアプリケーションに対する要求のルーターとして機能するターゲットとして、Web サーバーを指定します。各 Web サーバーのプラグイン構成ファイル (plugin-cfg.xml) は、経由して経路指定されるアプリケーションに基づいて生成されます。

クラスターおよびサーバー:

WebSphere:cell=IM8213Node01Cell,node=IM8213Node01,server=server1 適用

除去 更新 ファイルの除去 ファイルのエクスポート

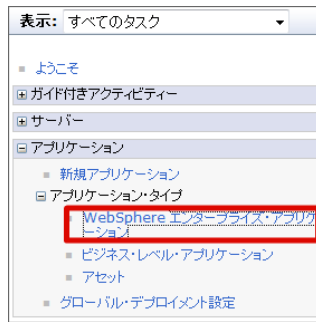
選択	モジュール	URI	モジュールタイプ	サーバー
☐	imart_war	imart_war.WEB-INF/web.xml	Web モジュール	WebSphere:cell=IM8213Node01Cell,node=IM8213Node01,server=server1

OK キャンセル

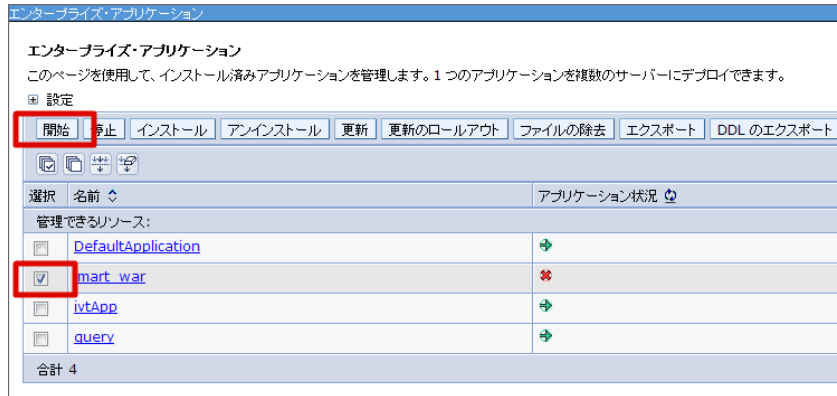
Webアプリケーションの開始

Webアプリケーションの開始を行います。

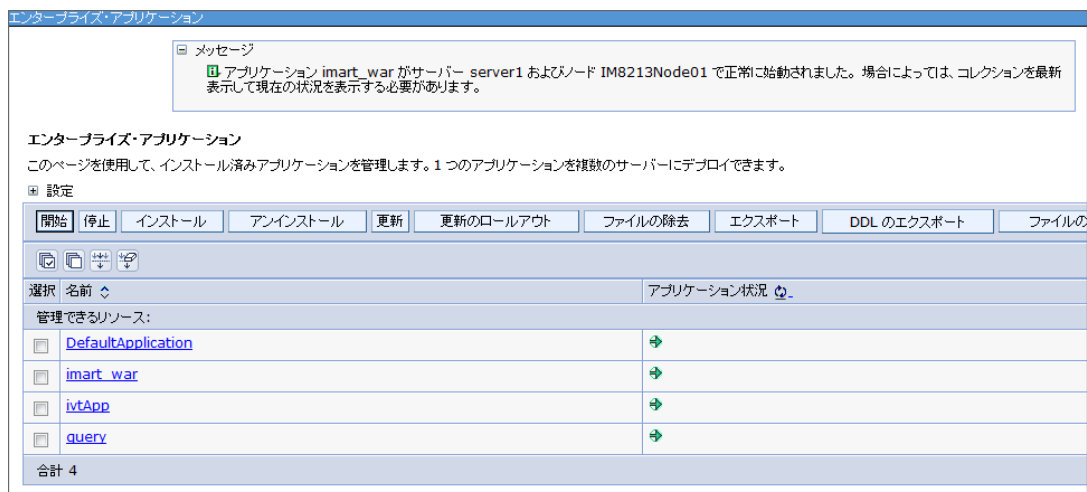
1. メニューから[アプリケーション]-[アプリケーション・タイプ]-[WebSphere エンタープライズ・アプリケーション] を選択します。



2. [imart_war] のチェックボックスをクリックし、[開始] ボタンをクリックします。



3. 下の画面のように正常終了のメッセージが表示されれば、Webアプリケーションの開始は完了です。



テナント環境の構築

intra-mart Accel Platform のテナント環境を構築します。

具体的な手順は「[テナント環境の構築](#)」を参照してください。

- [セットアップで困ったら...](#)
- [アップデート・パッチの適用](#)