



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

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
2016-12-01	第7版 下記を変更しました ▪ DB2に関する記述を削除
2019-04-01	第8版 下記を追加しました ▪ 「intra-mart Accel Platform 構成ファイルの作成」に Metro と OpenPortal WSRP の選択に関する注意事項を追加

本書の目的

本書では WebSphere Application Server 9.0.5 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 9.0.5 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。



注意

WebSphere Application Server を利用する場合、下記のモジュールを選択しないでください。起動時にエラーが発生します。

- ライブラリ > サードパーティ製ライブラリ > Metro
- ライブラリ > サードパーティ製ライブラリ > OpenPortal WSRP

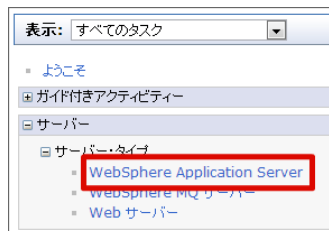
WebSphereの設定を行います。

WebSphere Application Server 9.0.5 のインストール手順については WebSphere Application Server 9.0.5 のマニュアルを参照してください。

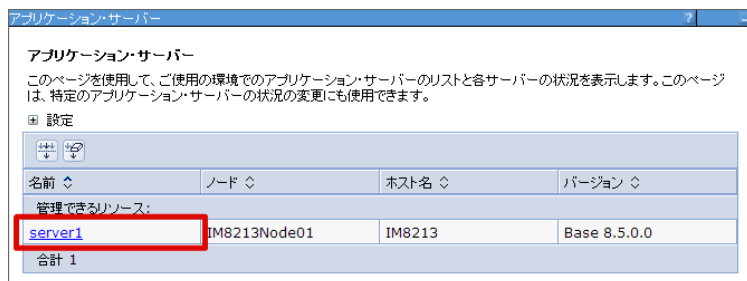
Java VM引数の設定

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。

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
 - サーバー・タイプ
 - WebSphere Application Server
 - WebSphere MQ サーバー
 - Web サーバー
- アプリケーション
- 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

セル= 、プロファイル=AppSrv01

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 > プロセス定義

このページを使用して、プロセス定義を構成します。プロセス定義で、プロセスを開始または初期化するために必要なコマンド行情報を定義します。

構成

一般プロパティ

追加プロパティ

- Java 仮想マシン
- 環境エントリ
- プロセスの実行
- プロセス・ログ
- ロギングおよびトレース

実行可能名

実行可能の引数

開始コマンド

開始コマンド引数

5. ヒープサイズを入力します。

☐ 冗長クラス・ロード

☐ 冗長ガーベッジ・コレクション

☐ 冗長 JNI

初期ヒープ・サイズ

1024 MB

最大ヒープ・サイズ

2048 MB

☐ HProf の実行

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

☐ デバッグ・モード

デバッグ引数

-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777

汎用 JVM 引数

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

実行可能 JAR ファイル名



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。

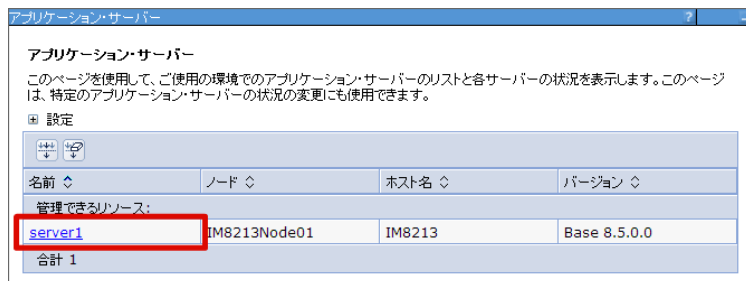
トランザクション・タイムアウトの設定

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。

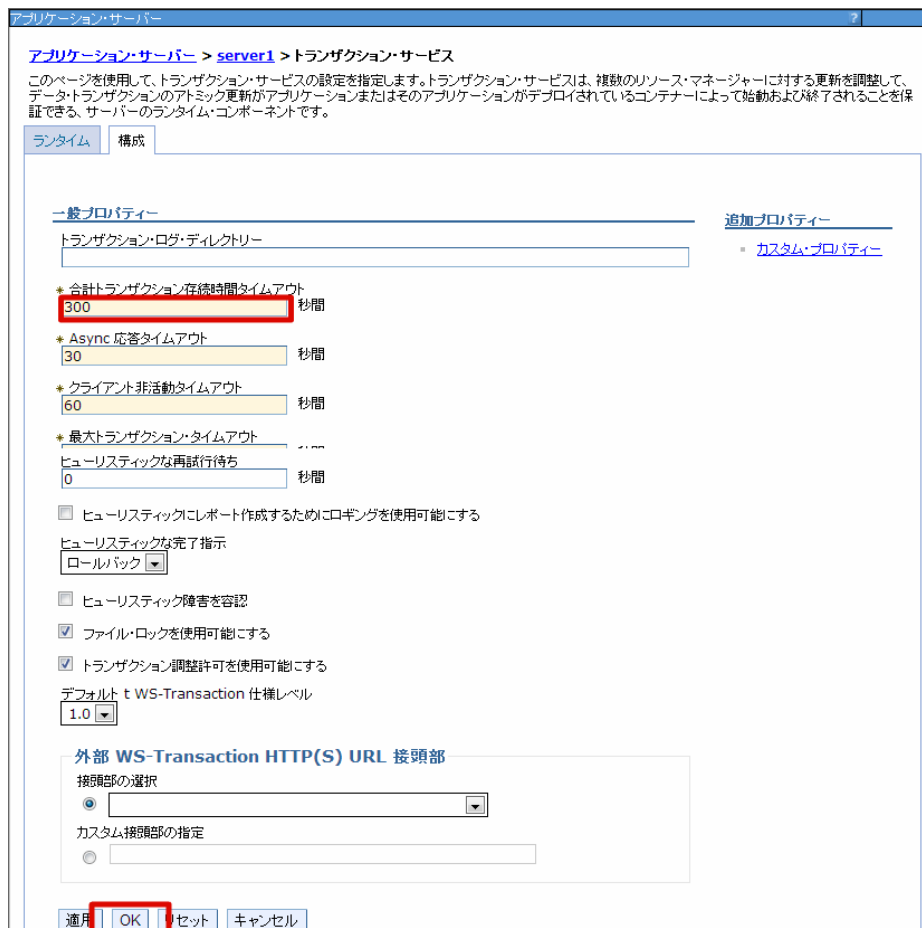
2. サーバーの名前を選択します。



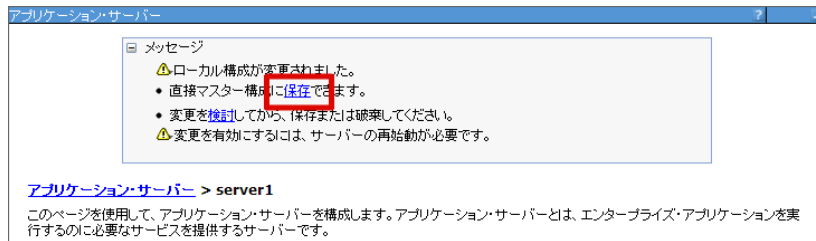
3. [コンテナ・サービス]-[トランザクション・サービス] を選択します。



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。



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。

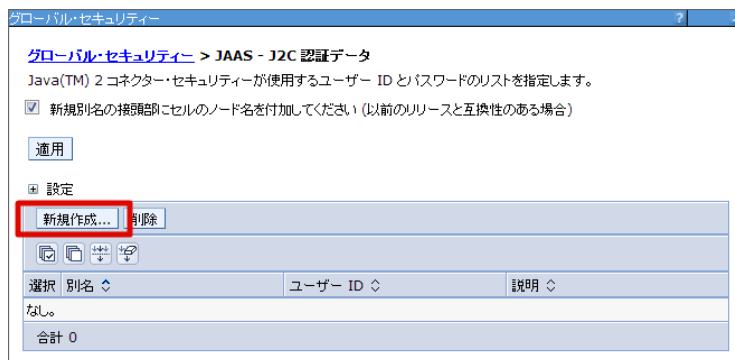
DBユーザ情報の設定

DBユーザ情報の設定を行います。

1. 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。



2. [新規作成] ボタンをクリックします。



3. 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:

☐	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

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

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。

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

Microsoft SQL Serverの設定

Microsoft SQL Server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

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

管理できるリソース:

選択	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

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

クラスパス:

適用

WebSphere 変数 `$ {MICROSOFT_JDBC_DRIVER_PATH}` として保存される "sqljdbc4.jar" のディレクトリー・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ー・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	<code>\$ {MICROSOFT_JDBC_DRIVER_PATH} /sqljdbc4.jar</code>
<code>\$ {MICROSOFT_JDBC_DRIVER_PATH}</code>	<code>c:/jdbc_driver</code>
ネイティブ・パス	<code>\$ {MICROSOFT_JDBC_DRIVER_NATIVEPATH}</code>
<code>\$ {MICROSOFT_JDBC_DRIVER_NATIVEPATH}</code>	
実装クラス名	<code>com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource</code>

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:	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

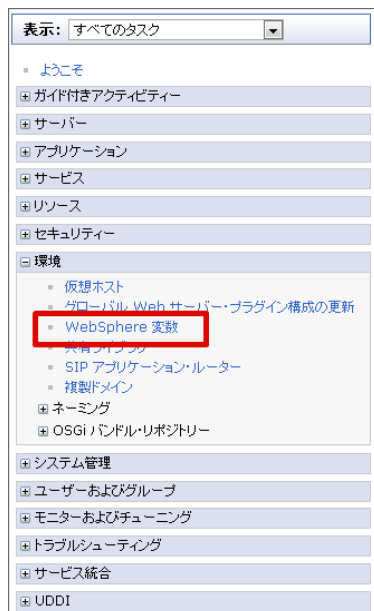
項目

- 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。

1. 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 > 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・ディレクトリーなどの、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。

WebSphere 変数

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・ディレクトリーなどの、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。

有効範囲: =すべての有効範囲

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](#)。

すべての有効範囲

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。

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

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
☐	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

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

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します。
- 変更を後述してから、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

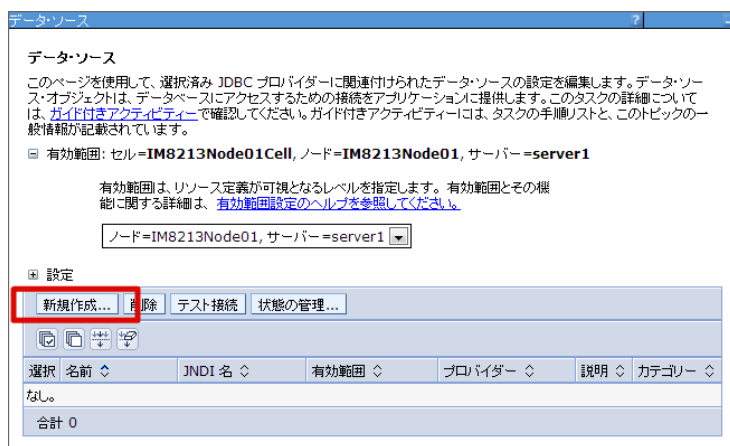
データ・ソースの登録

データソースの登録を行います。

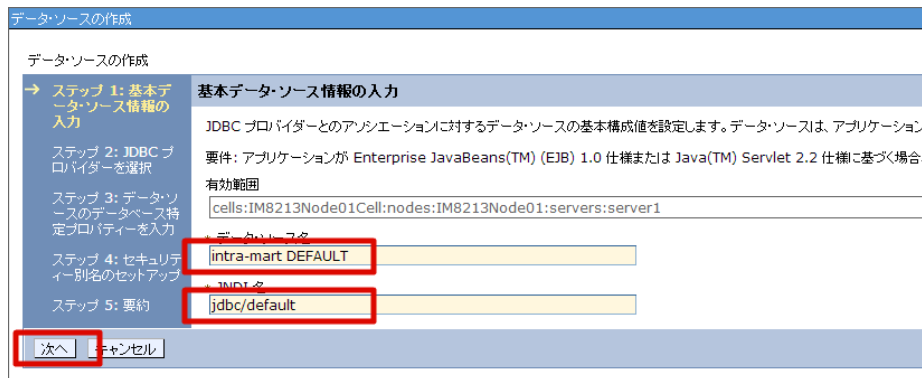
- メニューから[リソース]-[JDBC]-[データ・ソース]を選択します。



2. 「データ・ソース」画面で [新規作成] ボタンをクリックします。



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。



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 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

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

前へ 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、これらのデ

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

☒ コンテナー管理/パシスタンス (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、これらのデ

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

☒ コンテナー管理/パシスタンス (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

* データ・ストアのヘルパー・クラス名

com.ibm.websphere.rsadapter.GenericDataStoreHelper

☒ コンテナー管理/パシスタンス (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。」

6. [DB ユーザ情報の設定](#)で登録した[コンポーネント管理の認証別名]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 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](#)

前へ 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)

前へ 終了 キャンセル

8. [保存]をクリックします。Oracle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、サーバーの再起動が必要です。

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。

1. 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

2. [追加プロパティ]-[カスタム・プロパティ]を選択します。

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。

- PostgreSQLの場合

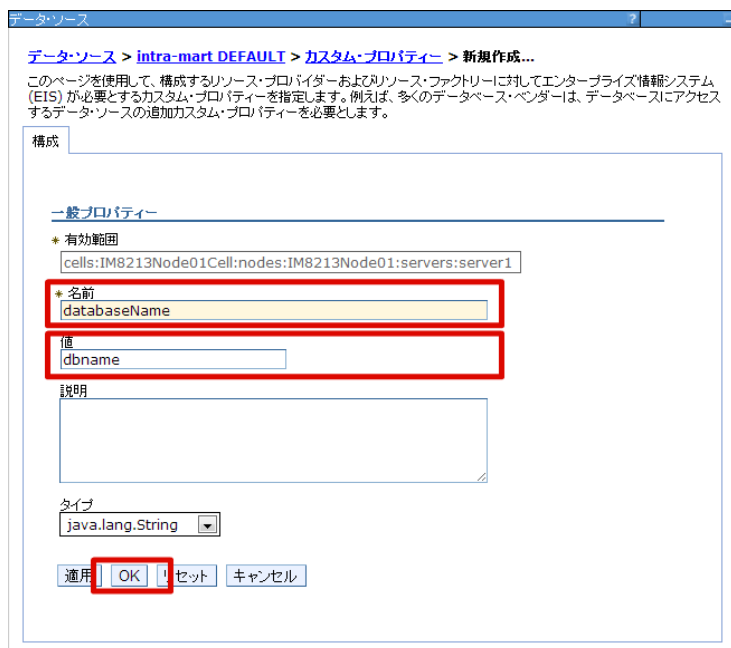
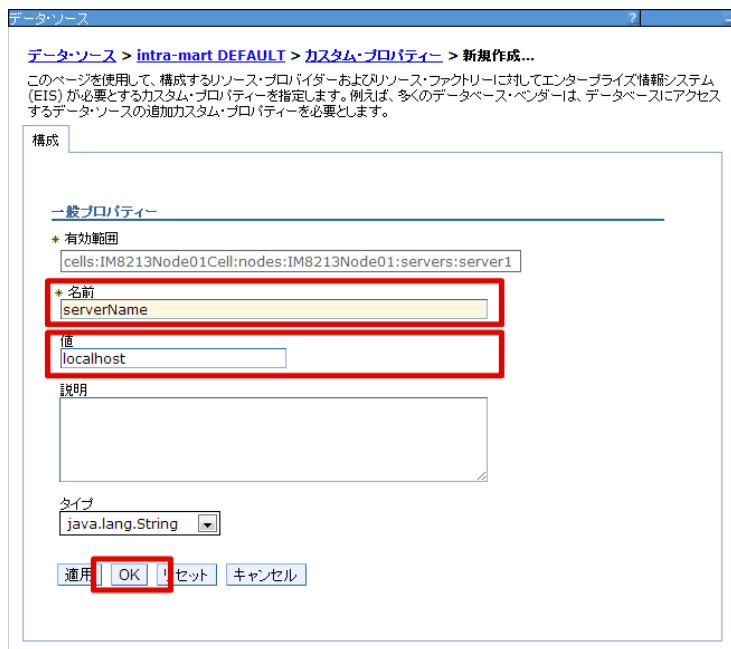
下記のプロパティを追加します。

- serverName
- databaseName
- portNumber

[新規作成] ボタンをクリックします。



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
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。

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

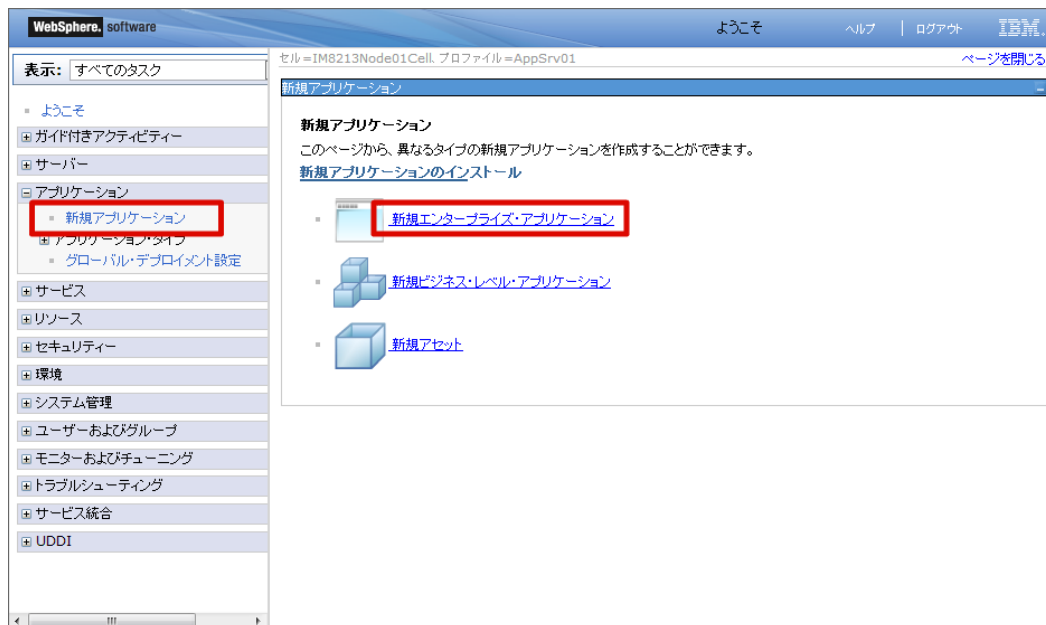
選択 名前 値 説明

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

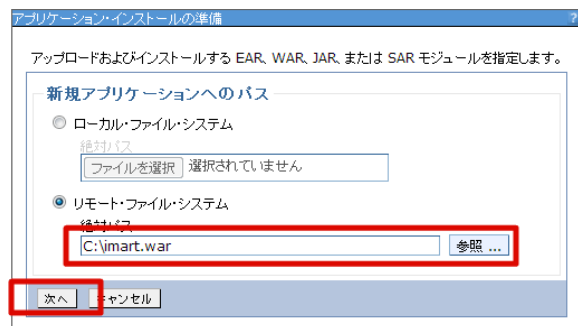
warファイルのデプロイを行います。

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。

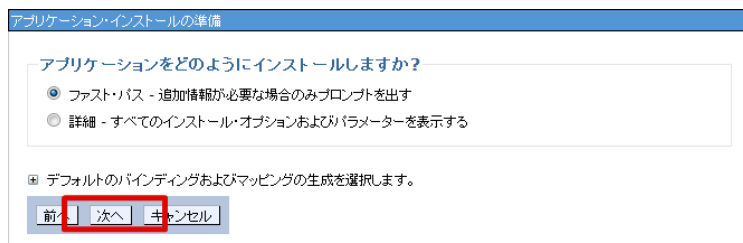


2. [新規アプリケーションへのパス]を設定します。

[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。



3. [次へ]ボタンをクリックします。



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

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

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

入力のエラー/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レクトリを許可する

☐ 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

非同期要求ディレクトリタイプ

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 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	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 ...

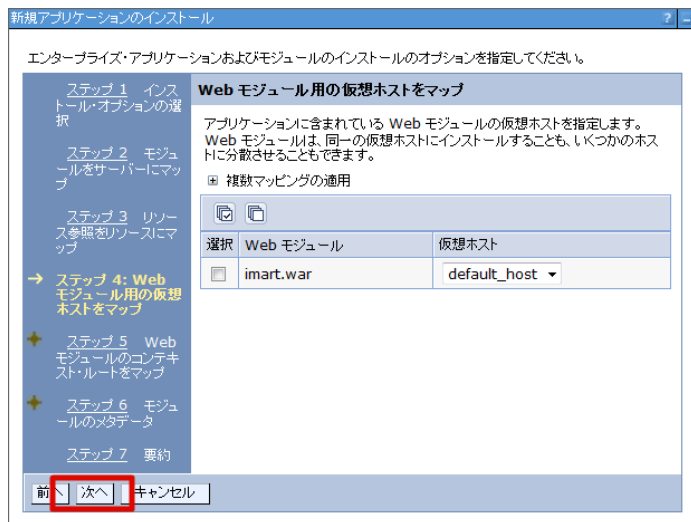
javax.sql.DataSource

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...

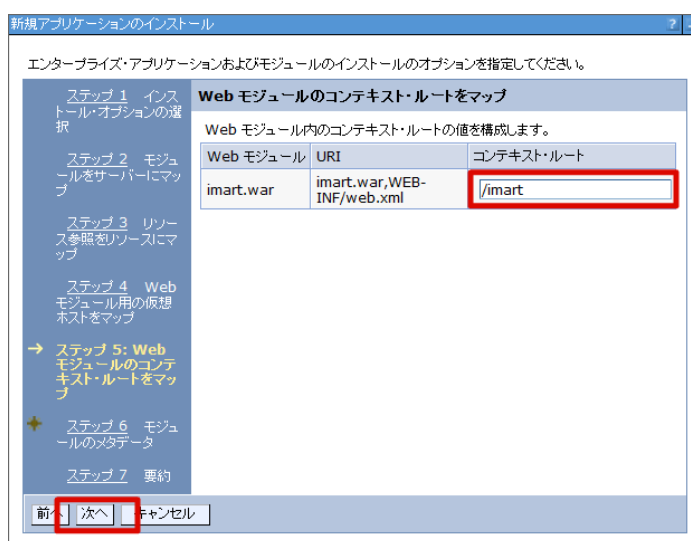
選択	モジュール	Bean	URI	リソース参照	ターゲット・リソース JNDI 名	ログイン構成
☐	imart.war		imart.war,WEB-INF/web.xml	jdbc/default	jdbc/default 参照 ...	リソース許可: コンテナー 認証方式: DefaultPrincipalMapping 認証データ入力: IM8213Node01/db_user

前へ 次へ キャンセル

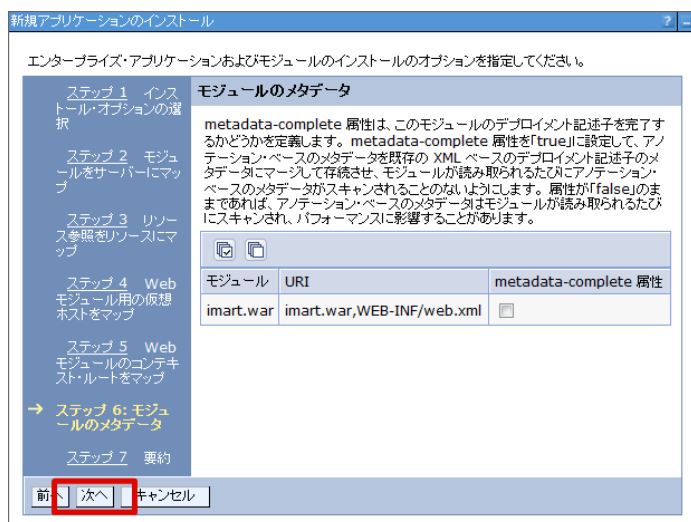
11. [次へ]ボタンをクリックします。



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。

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

要約

インストール・オプションの要約

オプション	値
JavaServer Pages ファイルのプリコンパイル	いいえ
アプリケーションをインストールするディレクトリー	
アプリケーションを配付する	はい
バイナリー構成の使用	いいえ
エンタープライズ Bean のデプロイ	いいえ
アプリケーション名	imart_war
リソース用の MBean を作成する	はい
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	いいえ
再ロード間隔 (秒)	
Web サービスをデプロイする	いいえ
入力のおつ/警告/失敗の検証	警告
プロセス組み込み構成	いいえ
ファイル・アクセス権	.*\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イスパッチを許可する	いいえ
リモート・リソースからの組み込みサービスを許可する	いいえ
ビジネス・レベル・アプリケーション名	
非同期要求デイスパッチ・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	いいえ
クライアント・モジュールをデプロイする	いいえ
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	いいえ
セル/ノード/サーバー	ここをクリックする

前 終了 キャンセル

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。

インストール中...

アプリケーションにエンタープライズ Bean がある場合、EJB デプロイメント処理に数分間かかる場合があります。処理が完了するまでは、構成を保存しないでください。

EJB デプロイメント処理に関する特定情報については、アプリケーションがデプロイされているデプロイメント・マネージャーまたはサーバーの SystemOut.log をチェックしてください。

ADMA5016t imart_war のインストールが開始されました。

ADMA5067t アプリケーション imart_war のリソース検証は正常に完了しました。

ADMA5058t アプリケーションとモジュールのバージョンが、デプロイメント・ターゲットのバージョンに対して検証されます。

ADMA5005t アプリケーション imart_war が WebSphere Application Server リポジトリに構成されます。

ADMA5005t アプリケーション imart_war が WebSphere Application Server リポジトリに構成されます。

ADMA5081t クライアント・モジュールのブートストラップ・アドレスが WebSphere Application Server リポジトリの中に構成されます。

ADMA5053t インストール済みオプション・パッケージのライブラリー参照が作成されます。

ADMA5005t アプリケーション imart_war が WebSphere Application Server リポジトリに構成されます。

ADMA5001t アプリケーション・バイナリー [D:\BM\WebSphere\AppServer\profiles\AppSrv01\workspace\cells\VM8213Node01Cell\applications\imart_war.ear\imart_war.ear] に保存されます。

ADMA5005t アプリケーション imart_war が WebSphere Application Server リポジトリに構成されます。

EJPPC0011t portlet.xml ファイルの検証が完了しました。

SECJ0400t アプリケーション imart_war が appContextIDForSecurity 情報で正常に更新されました。

ADMA5005t アプリケーション imart_war が WebSphere Application Server リポジトリに構成されます。

ADMA5005t アプリケーション imart_war が WebSphere Application Server リポジトリに構成されます。

ADMA5113t アクティベーション・プランが正常に作成されました。

ADMA5011t アプリケーション imart_war の一時ディレクトリーのクリーンアップが完了しました。

ADMA5013t 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。

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。

アプリケーションを開始するには、最初にマスター構成への変更を保存する必要があります。

ローカル構成が変更されました。

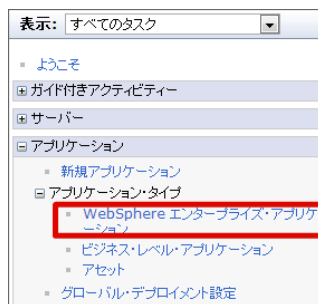
- 直接マスター構成に保存できます。
- 変更を[検証](#)してから、保存または破棄してください。

インストール済みのアプリケーションで作業するには、「アプリケーションの管理」リンクをクリックしてください。

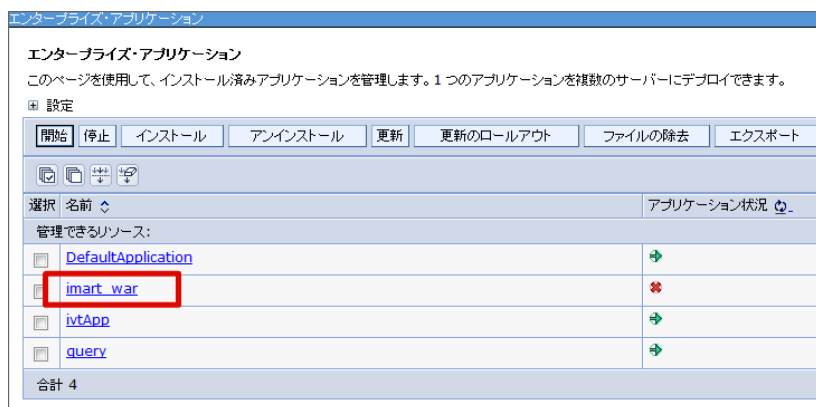
[アプリケーションの管理](#)

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、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。

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

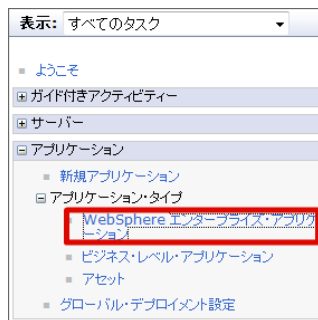
選択	モジュール	URI	モジュールタイプ	サーバー
☒	imart_war	imart_war,WEB-INF/web.xml	Web モジュール	WebSphere:cell=IM8213Node01Cell,node=IM8213Node01,server=server1

OK キャンセル

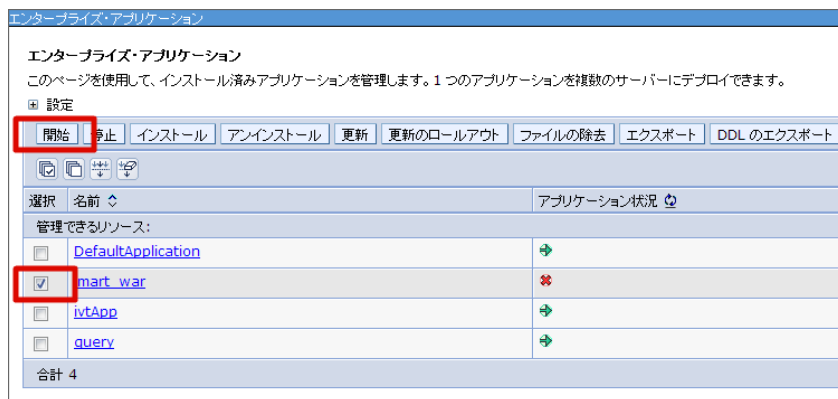
Webアプリケーションの開始

Webアプリケーションの開始を行います。

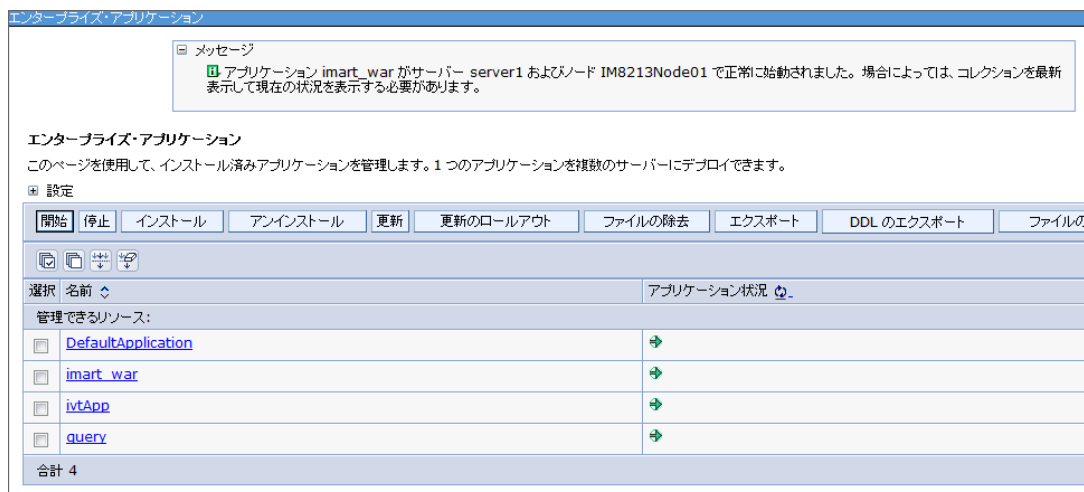
1. メニューから[アプリケーション]-[アプリケーション・タイプ]-[WebSphere エンタープライズ・アプリケーション]を選択します。



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。



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](#)