



目次

- 改訂情報
- はじめに
 - 本書の目的
 - 前提条件
 - 対象読者
 - 用語解説
- 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
- WebLogicの設定
 - Java ヒープサイズの設定
 - Java VM引数の設定
 - com.sun.mail.javax.mail.jar の差し替え
 - JAXBプロバイダの設定
 - タイムゾーンの設定
 - JTAの設定
 - データソースの設定
 - warファイルのデプロイ
- 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2版 下記を追加・変更しました <ul style="list-style-type: none">▪ 「Java VM引数の設定」を追加▪ 「タイムゾーンの設定」を追加▪ 「JAXB プロバイダの設定」を変更
2013-10-01	第3版 下記を追加・変更しました <ul style="list-style-type: none">▪ 「Java ヒープサイズの設定」を追加▪ 「DB2の設定」を変更▪ 「Microsoft SQL Serverの設定」を変更▪ 「Oracleの設定」を変更▪ 「PostgreSQLの設定」を変更
2014-01-01	第4版 下記を追加・変更しました <ul style="list-style-type: none">▪ 目次に「アップデート・パッチの適用」へのリンクを追加
2016-12-01	第5版 下記を追加・変更しました <ul style="list-style-type: none">▪ 「はじめに」に注意を追加▪ DB2に関する記述を削除
2017-04-01	第6版 下記を修正しました <ul style="list-style-type: none">▪ 「Java ヒープサイズの設定」の説明を修正
2018-04-01	第7版 下記を追加しました <ul style="list-style-type: none">▪ 「Oracleの設定」にコラムを追加
2019-04-01	第8版 下記を追加しました <ul style="list-style-type: none">▪ 「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
2020-08-01	第9版 下記を変更しました <ul style="list-style-type: none">▪ 「JAXB プロバイダの設定」のシステムプロパティの値を 2018 Winter(Urara) 以降において最適な値に修正
2022-06-01	第10版 下記を追加しました <ul style="list-style-type: none">▪ 「はじめに」の用語解説に %ORACLE_HOME% の説明を追加▪ 「com.sun.mail.javax.mail.jar の差し替え」を追加
2022-12-01	第11版 下記を変更しました <ul style="list-style-type: none">▪ Microsoft SQL Server に関する記述を削除
2023-04-01	第12版 下記を追加しました <ul style="list-style-type: none">▪ 「Java VM引数の設定」のコラムにカスタマーサクセスライセンスとプロキシ利用時に設定する JVM 引数を追加

本書の目的

本書では Oracle WebLogic Server 12c R2(12.2.1) 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。



注意

TERASOLUNA Server Framework for Java (5.x) for Accel Platform を利用する場合に制限事項があります。
詳細はリリースノートの「[制限事項](#)」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。

- Oracle WebLogic Server 12c R2(12.2.1) に intra-mart Accel Platform のセットアップを行われる方

用語解説

Oracle WebLogic Server 12c R2(12.2.1) のホームディレクトリを %WEBLOGIC_HOME% と略します。

%WEBLOGIC_HOME% が格納されているディレクトリを %ORACLE_HOME% と略します。

Apache HTTP Server をインストールしたディレクトリを %APACHE_HOME% と略します。

Storage として使用するディレクトリを %STORAGE_PATH% と略します。

Webサーバ利用時の静的コンテンツを配置するディレクトリを %WEB_PATH% と略します。

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

Oracle WebLogic 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

WebLogicの設定を行います。

Oracle WebLogic Server 12c R2(12.2.1) のインストール手順については Oracle WebLogic Server 12c R2(12.2.1) のマニュアルを参照してください。

Java ヒープサイズの設定

Java ヒープサイズの設定を行います。

1. WebLogic環境設定の実行ファイルを編集する。

Java ヒープサイズの設定はWebLogic環境設定の実行ファイルを編集してください。

Windowsの場合は **setDomainEnv.cmd**、Linuxの場合は **setDomainEnv.sh** です。

環境設定の実行ファイルをテキストエディタで開き、最少ヒープサイズ、最大ヒープサイズおよびMetaspace領域の初期値、Metaspace領域の最大値を設定します。

- Windowsの例

この例ではsetDomainEnv.cmdの「USER_MEM_ARGS」プロパティにインストール環境に応じたJavaヒープサイズ、Metaspace領域の設定をします。

```
...  
  
@REM IF USER_MEM_ARGS the environment variable is set, use it to override ALL MEM_ARGS values  
set USER_MEM_ARGS=-Xms1024m -Xmx2048m -XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=512m  
  
...
```

- Linuxの例

この例ではsetDomainEnv.shの「USER_MEM_ARGS」プロパティにインストール環境に応じたJavaヒープサイズ、Metaspace領域の設定をします。

```
...  
  
# IF USER_MEM_ARGS the environment variable is set, use it to override ALL MEM_ARGS values  
export USER_MEM_ARGS="-Xms1024m -Xmx2048m -XX:MetaspaceSize=256m -  
XX:MaxMetaspaceSize=512m"  
  
...
```

コラム

WebLogic Serverを開発モードで使用する際は、「USER_MEM_ARGS」プロパティに「-XX:CompileThreshold=8000」を追加して設定してください。

2. WebLogicを再起動します。

Java VM引数の設定

Java VM引数の設定を行います。

1. WebLogic起動用実行ファイルを編集する。

Java VM引数の設定はWebLogic起動用の実行ファイルを編集してください。Windowsの場合は

startWebLogic.cmd、Linuxの場合は **startWebLogic.sh** です。

起動用の実行ファイルをテキストエディタで開き、環境変数 **JAVA_OPTIONS** にJava VM引数の設定を追記します。

- Windowsの例

この例ではstartWebLogic.cmdの84行目にJava VM引数(-Duser.language=zh -Duser.country=CN)の設定を追記しています。

```

. . .

call "%DOMAIN_HOME%\bin\setDomainEnv.cmd" %*
set JAVA_OPTIONS=%JAVA_OPTIONS% -Duser.language=zh -Duser.country=CN
set SAVE_JAVA_OPTIONS=%JAVA_OPTIONS%

. . .

```

- Linuxの例
この例ではstartWebLogic.shの82行目にJava VM引数(-Duser.language=zh -Duser.country=CN)の設定を追記しています。

```

. . .

. ${DOMAIN_HOME}/bin/setDomainEnv.sh $*
JAVA_OPTIONS="${JAVA_OPTIONS} -Duser.language=zh -Duser.country=CN"
SAVE_JAVA_OPTIONS="${JAVA_OPTIONS}"

. . .

```

i コラム

カスタマーサクセスライセンスをご契約中の場合には、ライセンスポータルと通信してご契約内容の変更が自動反映されます。
WebLogic をインストールした環境からライセンスポータルへの通信にプロキシサーバを利用する場合は、次の JVM 引数の設定が必要です。

-DUseSunHttpHandle	WebLogic が HTTP 接続に Sun HTTP ハンドラを使用するか設定します。 プロキシサーバを利用する場合は、true を指定してください。
-Dhttps.proxyHost	プロキシサーバのホストURL
-Dhttps.proxyPort	プロキシサーバのポート番号
-Dhttps.proxyUser	プロキシサーバへの接続ユーザ
-Dhttps.proxyPassword	接続ユーザのパスワード

カスタマーサクセスライセンスについての詳細は、「[ライセンスの登録](#)」を参照してください。

2. WebLogicを再起動します。

com.sun.mail.javax.mail.jar の差し替え

1. Jakarta Mail 1.6.7 (jakarta.mail-1.6.7.jar) を以下のURLからダウンロードします。
<https://jakartaee.github.io/mail-api/> (English)

2. 以下のファイルを退避してください。

- Linux

```
%ORACLE_HOME%/oracle_common/modules/com.sun.mail.javax.mail.jar
```

- Windows

```
%ORACLE_HOME%\oracle_common\modules\com.sun.mail.javax.mail.jar
```

3. ダウンロードしたファイルを以下のパスに配置してください。

- Linux


```
%ORACLE_HOME%/oracle_common/modules/jakarta.mail-1.6.7.jar
```

- Windows

```
%ORACLE_HOME%\oracle_common\modules\jakarta.mail-1.6.7.jar
```

4. 配置したファイルのファイル名を退避を行った元のファイルと同じになるようにリネームしてください。

- Linux

```
mv %ORACLE_HOME%/oracle_common/modules/jakarta.mail-1.6.7.jar
%ORACLE_HOME%/oracle_common/modules/com.sun.mail.javax.mail.jar
```

- Windows

```
ren %ORACLE_HOME%\oracle_common\modules\jakarta.mail-1.6.7.jar
%ORACLE_HOME%\oracle_common\modules\com.sun.mail.javax.mail.jar
```

5. WebLogicを再起動します。

JAXBプロバイダの設定

JAXBプロバイダの設定を行います。

コラム

WebLogicで intra-mart Accel Platform を運用する場合、JAXBプロバイダの設定を行ってください。
JAXBプロバイダの設定を行わない場合、会社・組織などのエクスポートを行うと正しいデータが出力されません。

1. 下記のシステムプロパティを設定することで、JAXBプロバイダの設定が可能です。

システムプロパティ	値
com.sun.xml.ws.spi.db.BindingContextFactory	com.sun.xml.ws.db.glassfish.JAXBRIContextFactory
javax.xml.bind.JAXBContextFactory	com.sun.xml.bind.v2.ContextFactory

システムプロパティの指定はJava VM引数で指定します。上記のシステムプロパティを設定するには以下のJava VM引数に以下の設定を追記します。

```
-Dcom.sun.xml.ws.spi.db.BindingContextFactory=com.sun.xml.ws.db.glassfish.JAXBRIContextFactory -
Djavax.xml.bind.JAXBContextFactory=com.sun.xml.bind.v2.ContextFactory
```

注意

システムプロパティ `javax.xml.bind.JAXBContextFactory` は 2018 Winter 以降において有効なプロパティです。

2018 Summer 以前をご利用の場合は、システムプロパティ `javax.xml.bind.JAXBContextFactory` を `javax.xml.bind.JAXBContext` と指定してください。

```
-Dcom.sun.xml.ws.spi.db.BindingContextFactory=com.sun.xml.ws.db.glassfish.JAXBRIContextFactory
-Djavax.xml.bind.JAXBContext=com.sun.xml.bind.v2.ContextFactory
```

Java VM引数の設定方法については「[Java VM引数の設定](#)」を参照してください。

タイムゾーンの設定

コラム

タイムゾーンの初期設定は、Java VM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

こ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。

1. システムプロパティにタイムゾーンを設定します。

タイムゾーンはシステムプロパティ「user.timezone」で指定します。
システムプロパティの指定はJava VM 引数で行います。例えば、タイムゾーンを UTC に設定する場合はJava VM 引数に以下を追記します。

```
-Duser.timezone=UTC
```

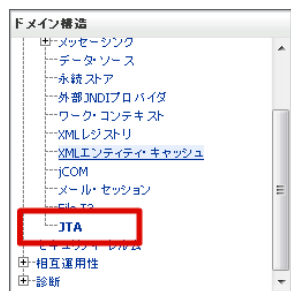
Java VM引数の設定方法については「[Java VM引数の設定](#)」を参照してください。

JTAの設定

JTA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、JTAのタイムアウト時間を変更します。

1. 左メニューの [サービス]-[JTA] を選択します。



2. [タイムアウト] を変更し、[保存]をクリックします。これで、JTAの設定は終わりです。

base_domainの設定

権威 監視 制御 セキュリティ Webサービスセキュリティ ノート

全般 JTA JPA EJB Webアプリケーション ログング ログ・フィルタ

保存

このページでは、このWebLogic ServerのメインのJavaランザクションAPI (JTA)構成を定義します。

タイムアウト:	<input type="text" value="300"/>	2フェーズ・コミット・トランザクションの第1フェーズで、アクティブなトランザクションが存続できる最大秒数を指定します。指定の時間が経過すると、トランザクションは自動的にロールバックされます。 詳細...
障害タイムアウト:	<input type="text" value="86400"/>	トランザクション・マネージャが、2フェーズ・コミット・トランザクションの第2フェーズを完了しようとして存続する最大秒数を指定します。 詳細...
beforeCompletionの反復上限:	<input type="text" value="10"/>	トランザクション・マネージャが、このWebLogic Serverのメインに対してbeforeCompletion同期化コールバックを実行するサイクルの最大数。 詳細...
最大トランザクション数:	<input type="text" value="10000"/>	このWebLogic Serverのメインのサーバーで許可される同時進行トランザクションの最大数。 詳細...
一差名の最大数:	<input type="text" value="1000"/>	保持する統計の対象となる一差のトランザクション名の最大数。 詳細...
チェックポイント間隔:	<input type="text" value="300"/>	トランザクション・マネージャが、新しいトランザクション・ログ・ファイルを作成し、すべての古いトランザクション・ログ・ファイルを確認してそれらを削除できるかどうかを確認する間隔。 詳細...
<input checked="" type="checkbox"/> ヒューリスティックの無視		トランザクション・マネージャがトランザクションのヒューリスティックな終了に対して自動的にXA Resource forget処理を実行するかどうかを指定します。 詳細...
リソース登録解除の猶予期間:	<input type="text" value="30"/>	トランザクション・マネージャが、リソースの関与するトランザクションの完了をリソースが登録解除されるまで待機する秒数。この猶予期間の設定は、登録解除されたリソース(アプリケーションと一緒にパッケージされているJDBCデータソース・モジュールなど)が原因でトランザクションが破棄されるリスクを最小限に抑えるために役立ちます。 詳細...
<input checked="" type="checkbox"/> XA呼出しの並列実行		使用可能なスレッドがある場合にXA呼出しを並列に実行することを指定します。 詳細...
<input checked="" type="checkbox"/> 2フェーズ・コミットの有効化		2フェーズ・コミット・プロトコルを使用して2つ以上のリソース・マネージャ間でトランザクションを調整することを指定します。 詳細...

[詳細](#)

保存

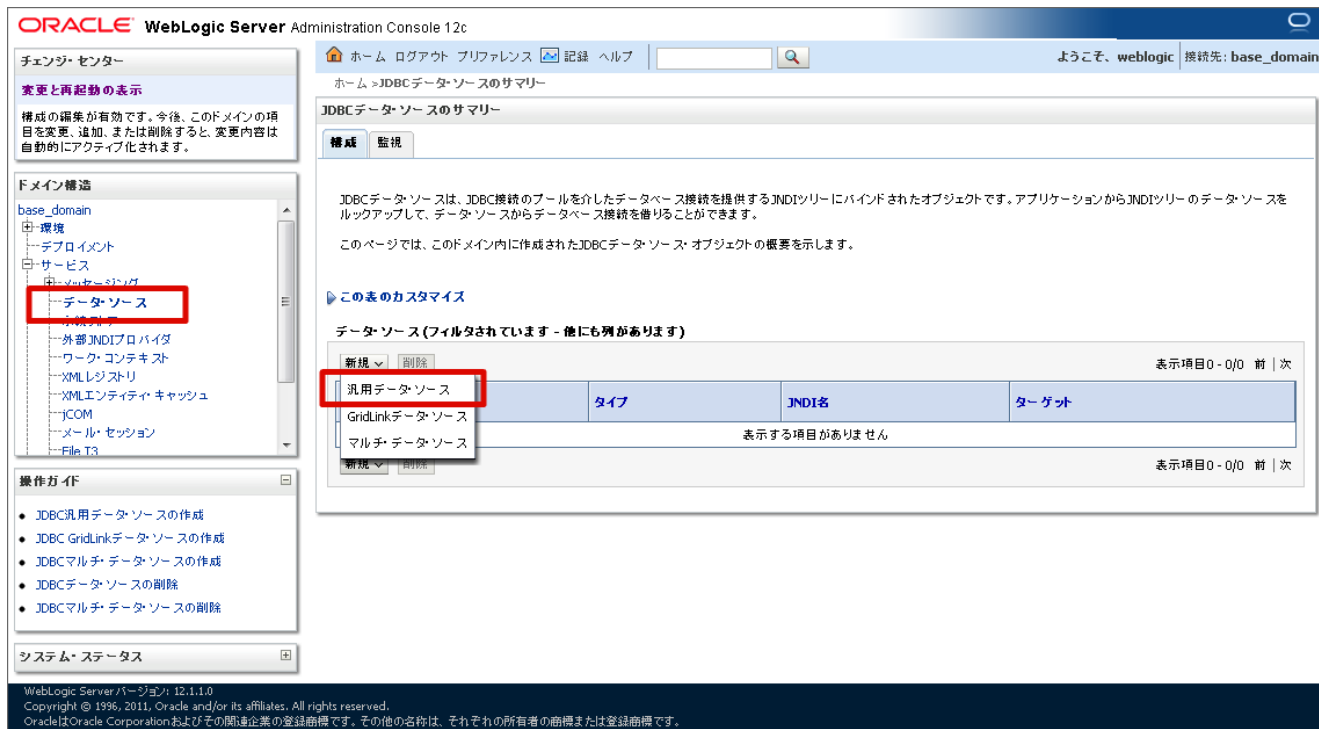
データソースの設定

データソースの設定を行います。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. 左メニューの [サービス]-[JDBC]-[データ ソース] を選択します。
右画面の[新規]ボタンをクリックし、[汎用データ・ソース]を選択します。



2. [名前]、[JNDI名]、[データベースのタイプ]を設定し、[次へ]ボタンをクリックします。

データベースのタイプ

Oracle



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。

3. [データベース・ドライバ]を設定し、[次へ]ボタンをクリックします。

データベース・ドライバ

Oracle's Driver (Thin) for Instance connections; Versions:
Any

新しいJDBCデータソースの作成

戻る 次 終了 取消

JDBCデータソースのプロパティ
次のプロパティは、新しいJDBCデータソースを識別するために使用されます。

データベースのタイプ: Oracle

データベース接続の作成に使用するデータベースドライバを指定してください。注意: *はドライバがOracle WebLogic Serverに明示的にサポートされていることを示します。

データベースドライバ: *Oracle's Driver (Thin) for Instance connections; Versions:9.0.1 and later

戻る 次 終了 取消



コラム

使用するDBがプラグブル・データベース (PDB) の場合は「**Oracle's Driver (Thin) for Service connections; Versions:Any**」を選択します。

4. [次へ]ボタンをクリックします。

新しいJDBCデータソースの作成

戻る 次 終了 取消

トランザクション オプション
非XA JDBCドライバを使用して新しいデータソースにデータベース接続を作成します。

このデータソースでグローバルトランザクションをサポートするかどうかを選択してください。サポートする場合は、このデータソースのトランザクションプロトコルを選択してください。

グローバルトランザクションのサポート

ロギング・ラスト・リソース (LLR)トランザクションの最適化を使用してデータソースからの非XA JDBC接続がグローバルトランザクションに参加できるようにする場合は、このオプションを選択します。「2フェーズ・コミットのエミュレート」のかわりに使用することをお勧めします。

ロギング・ラスト・リソース

JTAを使用してデータソースからの非XA JDBC接続がグローバルトランザクションへの参加をエミュレートできるようにする場合は、このオプションを選択します。このオプションは、ヒューリスティックな状況に耐えられるアプリケーションでのみ使用してください。

2フェーズ・コミットのエミュレート

1フェーズ・コミットトランザクション処理を使用してデータソースからの非XA JDBC接続がグローバルトランザクションに参加できるようにする場合は、このオプションを選択します。このオプションを有効にすると、他のリソースはグローバルトランザクションに参加できません。

1フェーズ・コミット

戻る 次 終了 取消

5. [データベース名]、[ホスト名]、[ポート]、[データベース・ユーザ名]、[パスワード]を設定し、[次へ]ボタンをクリックします。

新しいJDBCデータソースの作成

戻る 次 終了 取消

接続プロパティ
接続プロパティを定義します。

接続先のデータベースの名前を指定してください。

データベース名: orcl

データベース・サーバーの名前またはIPアドレスを指定してください。

ホスト名: localhost

データベースへの接続に使用するデータベース・サーバーのポートを指定してください。

ポート: 1521

データベース接続の作成に使用するデータベース・アカウントのユーザー名を指定してください。

データベース・ユーザー名: imart

データベース接続の作成に使用するデータベース・アカウントのパスワードを指定してください。

パスワード: *****

パスワードの確認: *****

戻る 次 終了 取消

6. [構成のテスト]を行い、接続テストに成功することを確認後、[次へ]ボタンをクリックします。
(接続テストに失敗する場合は、設定を再度見直してください)

新しいJDBCデータソースの作成

構成のテスト 戻る 次 終了 取消

データベース接続のテスト
データベースの可用性、および指定した接続プロパティをテストします。

接続プールでのデータベース接続の作成に使用するJDBCドライバ・クラスの完全パッケージ名を指定してください。
(このドライバ・クラスは、デプロイ先のいずれかのサーバーのクラスパスに含まれる必要があります。)

ドライバ・クラス名:

接続先データベースのURLを指定してください。使用するJDBCドライバによって、URLの書式が異なります。

URL:

データベース接続の作成に使用するデータベース・アカウントのユーザー名を指定してください。

データベース・ユーザー名:

データベース接続の作成に使用するデータベース・アカウントのパスワードを指定してください。
(注意: 安全なパスワード管理のために、パスワードは「プロパティフィールド」ではなく「パスワード」フィールドに入力してください。)

パスワード:

パスワードの確認:

データベース接続の作成時にJDBCドライバに渡すプロパティを指定してください。

プロパティ:

ドライバ・プロパティのセット。値は指定されたシステム・プロパティから実行時に導出されます。

システム・プロパティ:

データベース接続のテストに使用する表名またはSQL文を指定してください。

テスト対象の表名:

構成のテスト 戻る 次 終了 取消

7. JDBC データ ソースのデプロイ先をチェックし、[終了]ボタンをクリックします。
これで、JDBCデータ ソースの登録は終わりです。

新しいJDBCデータソースの作成

戻る 次 終了 取消

ターゲットの選択
新しいJDBCデータソースのデプロイ先として、1つまたは複数のターゲットを選択できます。ターゲットを選択しない場合でもデータソースは作成されますが、デプロイされません。その場合、後でデータソースをデプロイする必要があります。

サーバー

AdminServer

戻る 次 終了 取消

PostgreSQLの設定

PostgreSQLを使用する場合の設定を行います。

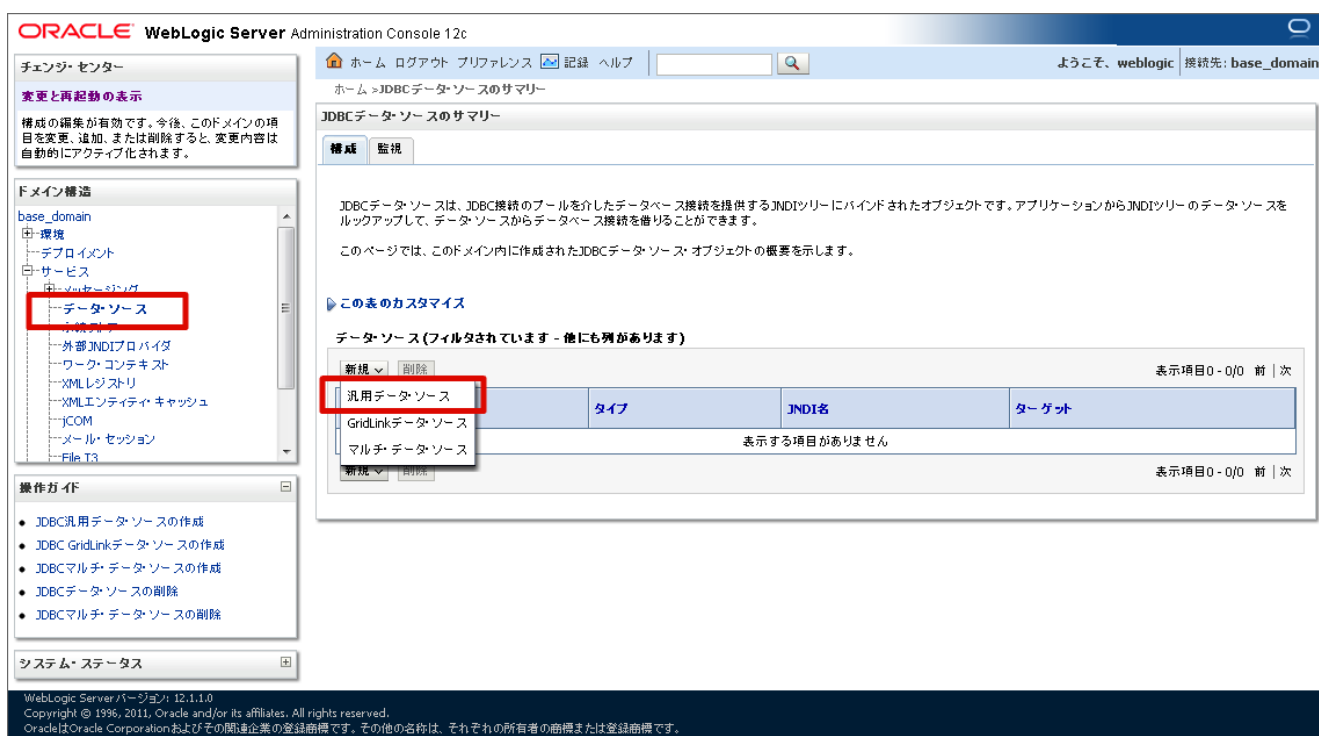
i コラム

PostgreSQLを使用する場合、予めJDBCドライバをWebLogicのクラスパスに通す必要があります。下記にその例を記載します。

1. JDBCドライバを **%WEBLOGIC_HOME%/server/lib** のディレクトリにコピーします。
2. Windowsの場合は **%WEBLOGIC_HOME%/common/bin/commEnv.cmd**、Linuxの場合は **commEnv.sh** を編集し、環境変数 **WEBLOGIC_CLASSPATH** にJDBCドライバのパスを追加します。下記はWindowsの例です。

```
set
WEBLOGIC_CLASSPATH=%JAVA_HOME%\lib\tools.jar;%WL_HOME%\server\lib\weblogic_sp.jar;%WL_HOME%\server\lib\all.jar;%ANT_CONTRIB%\lib\ant-contrib.jar;%WL_HOME%\server\lib\postgresql-9.1-902.jdbc4.jar
```

1. 左メニューの [サービス]-[JDBC]-[データ ソース] を選択します。
右画面の[新規]ボタンをクリックし、[汎用データ・ソース]を選択します。



2. [名前]、[JNDI名]、[データベースのタイプ]を設定し、[次へ]ボタンをクリックします。

データベースのタイプ

PostgreSQL

新しいJDBCデータソースの作成

戻る 次 終了 取消

JDBCデータソースのプロパティ
次のプロパティは、新しいJDBCデータソースを識別するために使用されます。
*は必須フィールドです

新しいJDBCデータソースの名前を指定してください。

*名前:

新しいJDBCデータソースのJNDI名を指定してください。

JNDI名:

データベースのタイプを選択してください。

データベースのタイプ:

戻る 次 終了 取消

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。

3. [データベース・ドライバ]を設定し、[次へ]ボタンをクリックします。

データベース・ドライバ

PostgreSQL's Driver (Type 4) Versions:
Any

新しいJDBCデータソースの作成

戻る 次 終了 取消

JDBCデータソースのプロパティ
次のプロパティは、新しいJDBCデータソースを識別するために使用されます。

データベースのタイプ: PostgreSQL

データベース接続の作成に使用するデータベース・ドライバを指定してください。注意: *はドライバがOracle WebLogic Serverで明示的にサポートされていることを示します。

データベース・ドライバ:

戻る 次 終了 取消

4. [次へ]ボタンをクリックします。

新しいJDBCデータソースの作成

戻る 次 終了 取消

トランザクション・オプション
非XA JDBCドライバを使用して新しいデータソースにデータベース接続を作成します。

このデータソースでグローバルトランザクションをサポートするかどうかを選択してください。サポートする場合は、このデータソースのトランザクション・プロトコルを選択してください。

グローバルトランザクションのサポート

ロギング・ラスト・リソース (LLR) トランザクションの最適化を使用してデータソースからの非XA JDBC接続がグローバルトランザクションに参加できるようにする場合は、このオプションを選択します。「2フェーズ・コミットのエミュレート」のかわりに使用することをお薦めします。

ロギング・ラスト・リソース

JTAを使用してデータソースからの非XA JDBC接続がグローバルトランザクションへの参加をエミュレートできるようにする場合は、このオプションを選択します。このオプションは、ヒューリスティックな状況に耐えられるアプリケーションでのみ使用してください。

2フェーズ・コミットのエミュレート

1フェーズ・コミットトランザクション処理を使用してデータソースからの非XA JDBC接続がグローバルトランザクションに参加できるようにする場合は、このオプションを選択します。このオプションを有効にすると、他のリソースはグローバルトランザクションに参加できません。

1フェーズ・コミット

戻る 次 終了 取消

5. [データベース名]、[ホスト名]、[ポート]、[データベース・ユーザ名]、[パスワード] を設定し、[次へ]ボタンをクリックします。

新しいJDBCデータソースの作成

戻る 次 終了 取消

接続プロパティ
接続プロパティを定義します。

接続先のデータベースの名前を指定してください。

データベース名:

データベース・サーバーの名前またはIPアドレスを指定してください。

ホスト名:

データベースへの接続に使用するデータベース・サーバーのポートを指定してください。

ポート:

データベース接続の作成に使用するデータベース・アカウントのユーザー名を指定してください。

データベース・ユーザー名:

データベース接続の作成に使用するデータベース・アカウントのパスワードを指定してください。

パスワード:

パスワードの確認:

戻る 次 終了 取消

6. [構成のテスト]を行い、接続テストに成功することを確認後、[次へ]ボタンをクリックします。
(接続テストに失敗する場合は、設定を再度見直してください)

新しいJDBCデータソースの作成

構成のテスト 戻る 次 終了 取消

データベース接続のテスト

データベースの可用性、および指定した接続プロパティをテストします。

接続プールでのデータベース接続の作成に使用するJDBCドライバ・クラスの完全パッケージ名を指定してください。
(このドライバ・クラスは、デプロイ先のいずれかのサーバーのクラスパスに含まれる必要があります。)

ドライバ・クラス名:

接続先データベースのURLを指定してください。使用するJDBCドライバによって、URLの書式が異なります。

URL:

データベース接続の作成に使用するデータベース・アカウントのユーザー名を指定してください。

データベース・ユーザー名:

データベース接続の作成に使用するデータベース・アカウントのパスワードを指定してください。
(注意: 安全なパスワード管理のために、パスワードは「プロパティフィールド」ではなく「パスワード」フィールドに入力してください)

パスワード:

パスワードの確認:

データベース接続の作成時にJDBCドライバに渡すプロパティを指定してください。

プロパティ:

ドライバ・プロパティのセット。値は指定されたシステム・プロパティから実行時に導出されます。

システム・プロパティ:

データベース接続のテストに使用する表名またはSQL文を指定してください。

テスト対象の表名:

構成のテスト 戻る 次 終了 取消

7. JDBC データ ソースのデプロイ先をチェックし、[終了]ボタンをクリックします。
これで、JDBCデータ ソースの登録は終わりです。

新しいJDBCデータソースの作成

戻る 次 終了 取消

ターゲットの選択

新しいJDBCデータソースのデプロイ先として、1つまたは複数のターゲットを選択できます。ターゲットを選択しない場合でもデータソースは作成されますが、デプロイされません。その場合、後でデータソースをデプロイする必要があります。

サーバー

AdminServer

戻る 次 終了 取消

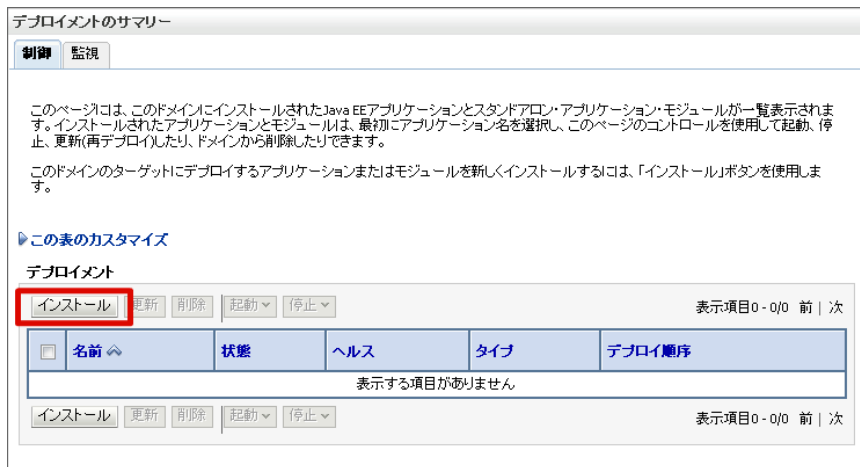
warファイルのデプロイ

warファイルのデプロイを行います。

1. 左メニューの [デプロイメント] を選択します。



2. [インストール]ボタンをクリックします。



3. [パス]にwarファイルのフルパスを入力し、[次]ボタンをクリックします。



4. [次]ボタンをクリックします。



5. [次]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・アシスタント

戻る 次 終了 取消

オプション設定
これらの設定は、変更することもデフォルトを受け入れることもできます

— 全般 —
このデプロイメントの名前を指定してください。

名前:

— セキュリティ —
このアプリケーションに使用するセキュリティ・モデルを指定してください。

- DDのみ: デプロイメント・ディスクリプタ内に定義されたロールとポリシーのみを使用します。
- カスタム・ロール: 管理コンソール内に定義されたロールを使用します。デプロイメント・ディスクリプタ内に定義されたポリシーを使用します。
- カスタム・ロールおよびポリシー: 管理コンソール内に定義されたロールとポリシーのみを使用します。
- 詳細: レルムの構成ページで構成したカスタム・モデルを使用します。

— ソースのアクセス可能性 —
ソース・ファイルをどのようにアクセス可能にするかを指定してください。

- デプロイメントのターゲットに定義されるデフォルトを使用

推奨の選択。

- すべてのターゲットにこのアプリケーションをコピーする

デプロイ時に、アプリケーションが割り当てられる管理ターゲット・サーバーにファイルが自動的にコピーされます。

- デプロイメントを次の場所からアクセス可能にする

場所:

すべてのターゲットからアクセスされる、このアプリケーションのファイルの場所を指定してください。これは通常は共有ディレクトリです。この場所にアプリケーションのファイルが存在し、すべてのターゲットからこの場所にアクセスできる必要があります。

戻る 次 終了 取消

6. [終了]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・アシスタント

戻る 次 終了 取消

選択項目を確認して「終了」をクリック
デプロイメントを完了する口は、「終了」をクリックします。完了するまでしばらくかかる場合があります。

— 追加構成 —
このアプリケーションが正常に機能する口は、追加構成が必要な場合があります。このアシスタントの完了後に、このアプリケーションの構成を確認しますか。

- はい、デプロイメントの構成画面に移動します。
- いいえ、後で構成を確認します。

— サマリー —

デプロイメント: D:\work\imart.war

名前: imart

ステージング・モード: 選択したターゲットに定義されるデフォルトを使用

セキュリティ・モデル: DDのみ: デプロイメント・ディスクリプタ内に定義されたロールとポリシーのみを使用します。

ターゲットのサマリー

コンポーネント	ターゲット
imart	AdminServer

戻る 次 終了 取消

7. [保存]ボタンをクリックします。これで、warファイルのデプロイは終わりです。

imartの設定

[概要](#)
[デプロイメント・プラン](#)
[構成](#)
[セキュリティ](#)
[ターゲット](#)
[制御](#)
[テスト](#)
[監視](#)
[ノート](#)

保存

このページでは、Webアプリケーションのインストール済み構成を表示します。

名前:	imart	このアプリケーション・デプロイメントの名前。 詳細...
コンテキスト・ルート:	/imart	サーブレットがこのWebアプリケーションを探すときに使用する、特定のパス。 詳細...
パス:	D:\work\imart.war	管理サーバー上のデプロイ可能なユニットのソースへのパス。 詳細...
デプロイメント・プラン:	(プラン未指定)	管理サーバー上のデプロイメント・プランドキュメントへのパス。 詳細...
ステージング・モード:	(未指定)	アプリケーションの準備段階に、管理サーバーのソースから管理対象サーバーのステージング領域にアプリケーションのファイルがコピーされるかどうかを指定するモード。 詳細...
セキュリティ・モデル:	DDOnly	セキュリティ・モデルでは、デプロイメントを保護する方法を指定します。 詳細...
デプロイ順序:	<input type="text" value="100"/>	起動中にこのユニットがデプロイされたときを表す数値で、サーバー上の他のデプロイ可能ユニットに対する相対的な値。 詳細...
デプロイメント・プリンシパル名:	<input type="text"/>	起動および停止中にファイルまたはアーカイブをデプロイするときに使用されるプリンシパルを示す文字列値。このプリンシパルは、ApplicationLifecycleListenerなどのインタフェース用のアプリケーション・コードを呼び出すときに、現在のサブジェクトを設定するために使用されます。プリンシパル名が指定されていない場合、匿名プリンシパルが使用されます。 詳細...

保存

モジュールおよびコンポーネント

表示項目 1 - 1/1 前 | 次

名前	タイプ
<input type="checkbox"/> imart	Webアプリケーション
<input type="checkbox"/> Webサービス	
表示項目なし	

表示項目 1 - 1/1 前 | 次

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](#)