

# WEBSPHERE を使用した JEE クラスター上の ADOBE EXPERIENCE MANAGER FORMS の設定

## 法律上の注意

法律上の注意については、[http://help.adobe.com/ja\\_JP/legalnotices/index.html](http://help.adobe.com/ja_JP/legalnotices/index.html) を参照してください。

# 目次

<b>第 1 章：WebSphere Application Server クラスターの作成</b>	
1.1 インストールの準備 .....	1
1.2 WebSphere Network Deployment ソフトウェアのインストール .....	2
1.3 WebSphere クラスターの作成と設定 .....	3
1.4 WebSphere Application Server クラスターのテスト .....	11
1.5 次の手順 .....	12
<b>第 2 章：AEM forms モジュールのインストール</b>	
<b>第 3 章：AEM Forms をデプロイするための設定</b>	
<b>第 4 章：WebSphere クラスターの手動設定</b>	
4.1 ディレクトリ権限 .....	15
4.2 WebSphere Application Server インスタンスの設定 .....	16
4.3 AEM forms データベース接続の設定 .....	19
4.4 次の手順 .....	29
<b>第 5 章：デプロイメント後のタスク</b>	
<b>第 6 章：ロードバランシングの設定</b>	
6.1 インストールの準備 .....	31
6.2 Web サーバーのインストール .....	31
6.3 Web サーバープラグインのインストール .....	32
<b>第 7 章：付録 - デプロイヤーのヒープサイズの拡大 (WebSphere 用)</b>	

# 第 1 章：WebSphere Application Server クラスターの作成

WebSphere クラスターを作成するには、WebSphere Application Server Network Deployment ソフトウェアをインストールする必要があります。次のタスクを実行します。

- クラスター内のすべてのコンピューターが正しく準備されていることを確認します（1 ページの「[1.1 インストールの準備](#)」を参照）。
- WebSphere Application Server Network Deployment ソフトウェアをインストールします（2 ページの「[1.2 WebSphere Network Deployment ソフトウェアのインストール](#)」を参照）。
- WebSphere Application Server クラスターを作成します（3 ページの「[1.3 WebSphere クラスターの作成と設定](#)」を参照）。
- WebSphere Application Server クラスターの設定をテストします（11 ページの「[1.4 WebSphere Application Server クラスターのテスト](#)」を参照）。

## 1.1 インストールの準備

WebSphere Application Server をクラスター内のコンピューターにインストールする前に、システムが以下の構成要件を満たしていることを確認してください。

**ディスク容量：**アプリケーションサーバーをインストールするパーティションに 10 GB 以上の空きディスク容量があることを確認します。製品のインストールに必要な容量に加えて、環境変数 TEMP または TMP が、最低 500 MB の空き容量がある有効な一時ディレクトリを指している必要があります。ダウンロード可能な実行ファイルに約 500 MB、イメージの展開用にさらに 1.0 GB 必要です。

**IP アドレス設定：**すべてのコンピューターに、単一の DNS で管理される固定 IP アドレスが必要です。

**IP マルチキャスト：**すべてのコンピューターが、IP マルチキャストパケットの転送を完全にサポートしている必要があります。つまり、すべてのルーターやその他のトンネル用テクノロジーが、クラスターサーバーインスタンスにマルチキャストメッセージを転送するように設定されている必要があります。ネットワーク遅延は、最低でも、ほとんどのマルチキャストメッセージの最終到達先に 200 ~ 300 ミリ秒以内に到達するレベルにあることが必要です。また、クラスターのマルチキャスト time-to-live (TTL) 値は、マルチキャストパケットが最終到達先に届く前にルーターによって破棄されることのないよう、十分に大きい値である必要があります。

**バージョン：**クラスター内のすべてのコンピューターは、WebSphere Application Server ソフトウェアの同一バージョンおよび同一サービスパックを使用している必要があります。

**水平クラスタリング：**水平クラスター設置では、（つまり WebSphere Application Server のインスタンスが別々のコンピューターにインストールされている場合は）すべてのコンピューターが同一のネットワークサブネット上にあり、コンピューターの時間が同期されている必要があります（『[AEM forms のインストールの準備（サーバークラスター）](#)』を参照）。

**アカウント権限：**(Windows) WebSphere Application Server は、管理者権限を持つユーザーアカウントでインストールおよび実行する必要があります。

**共有ネットワークドライブ：**クラスター内のすべてのコンピューターに読み取りと書き込み権限がある、安全な共有ネットワークドライブを作成しておく必要があります（『[AEM forms のインストールの準備（サーバークラスター）](#)』を参照）。

クラスターのすべてのシステムのクロックが共通のタイムサーバーと同期されている可能性があります。Windows ドメインでは、クロックの同期は自動的に行われます。Windows 以外のシステムでは、Network Time Protocol を設定する必要があります。

## 1.2 WebSphere Network Deployment ソフトウェアのインストール

クラスター内の各ノードに WebSphere Application Server Network Deployment ソフトウェアをインストールして、WebSphere Deployment Manager およびアプリケーションサーバーインスタンスを作成する必要があります。

WebSphere Deployment Manager を使用して、WebSphere Application Server クラスターを管理できます。WebSphere Application Server Network Deployment は、専用の管理コンピューターにインストールできるほか、クラスター内で Deployment Manager をインストールして実行するのに十分な容量のあるノードにもインストールできます (WebSphere Application Server サイトを参照)。

### 1.2.1 WebSphere Network Deployment ソフトウェアのインストールモード

WebSphere Network Deployment ソフトウェアは、次のいずれかの方法でインストールできます。

- プロファイルを作成せずに WebSphere Network Deployment ソフトウェアをインストールするには、**WebSphere Application Server Environments** 画面で「**None**」を選択します。manageprofiles スクリプトを使用して、後から Deployment Manager またはアプリケーションサーバープロファイルを作成することができます。3 ページの「[1.3.1 WebSphere プロファイルの作成](#)」を参照してください。
- WebSphere Network Deployment ソフトウェアを単一のアプリケーションサーバープロファイルと共にインストールするには、**WebSphere Application Server Environments** 画面で「**Application server**」を選択します。
- WebSphere Network Deployment ソフトウェアを Deployment Manager プロファイルと共にインストールするには、**WebSphere Application Server Environments** 画面で「**Management**」を選択します。
- WebSphere Network Deployment ソフトウェアを Deployment Manager および管理対象のノードのアプリケーションサーバープロファイルを含むセルと共にインストールするには、**WebSphere Application Server Environments** 画面で「**Cell (Deployment Manager and a Managed Node)**」オプションを選択します。

インストール時にこのオプションを選択すると、後から手動でプロファイルを作成する手間を省くことができます。

WebSphere Application Server ソフトウェアのインストールについては、[WebSphere Application Server](#) サイトを参照してください。

**注意：**アプリケーションサーバーのみをホストするノードに WebSphere Network Deployment ソフトウェアをインストールする場合は、**WebSphere Application Server Environments** 画面で「**Application server**」オプションを選択します。

**注意：**manageprofiles スクリプトを使用すると、前述のいずれかの方法で WebSphere Network Deployment ソフトウェアをインストールした後に、Deployment Manager またはアプリケーションサーバープロファイルを作成することができます。

## 1.3 WebSphere クラスターの作成と設定

以下のタスクを実行して、WebSphere Application Server クラスターを設定します。

- WebSphere Deployment Manager および WebSphere Application Server プロファイルを作成します (3 ページの「[1.3.1 WebSphere プロファイルの作成](#)」を参照)。
- ノードを Deployment Manager に統合します (6 ページの「[1.3.3 WebSphere Application Server プロファイルの統合](#)」を参照)。
- クラスターを作成します (8 ページの「[1.3.4 WebSphere クラスターの作成](#)」を参照)。
- SOAP 接続タイムアウトを変更します (10 ページの「[1.3.5 SOAP 接続タイムアウト設定の変更](#)」を参照)。

プロファイルを作成すると、管理コンソールもサーバーも含まない空のノードが作成されます。これらのノードを統合した後に、Deployment Manager を使用して、これらのノードを使用するサーバーのクラスターを作成します。

### 1.3.1 WebSphere プロファイルの作成

2 ページの「[1.2.1 WebSphere Network Deployment ソフトウェアのインストールモード](#)」に記載されている方法のいずれかを使用して WebSphere Network Deployment ソフトウェアをインストールすると、様々な種類の WebSphere プロファイルを作成できます。WebSphere Deployment Manager 用と WebSphere Application Server インスタンス用のプロファイルを作成します。

WebSphere Application Server Network Deployment ソフトウェアをインストールするときに「**Cell (Deployment Manager and a Managed Node)**」オプション (WebSphere Application Server environments 画面) を選択した場合は、Deployment Manager と管理対象のアプリケーションサーバーのプロファイルが自動的に作成されています。

#### 1.3.1.1 64 ビット WebSphere Application Server 用 WebSphere プロファイルの作成

64 ビットの WebSphere Application Server でプロファイルを作成するには、WebSphere の manageprofiles スクリプト (Windows の場合は manageprofiles.bat、Linux または UNIX の場合は manageprofiles.sh) を使用します。

WebSphere Deployment Manager のホストとして選択したノード上で、Deployment Manager のプロファイルを作成します。このプロファイルは、WebSphere Administrative Console が含まれ、クラスターのノードの統合先となるセルのホストの役割も果たします。

また、クラスターを構成する WebSphere Application Server インスタンスに関しても、それぞれプロファイルを作成する必要があります。

#### JDK 1.7 を使用するための WebSphere および WebSphere の設定

プロファイルの作成前に、次の手順を実行して、WebSphere サーバーおよび新しく作成したプロファイルが JDK 1.7 を使用することを確認します。

<WAS\_HOME>\AppServer\bin ディレクトリから次のコマンドを実行します。

- 1 使用できる SDK の一覧表示：

```
managesdk -listAvailable
```

- 2 デフォルト SDK の SDK 7.0 への変更：

```
managesdk -setCommandDefault -sdkname 1.7_64
```

- 3 SDK 7.0 を使用するための新しいプロファイルの設定：

```
managesdk -setNewProfileDefault -sdkname 1.7_64
```

既にプロファイルを作成済みの場合のみ、以下に一覧表示したコマンドを実行してください。

```
managesdk -listAvailable [-verbose]
managesdk -listEnabledProfile -profileName AppSrv01 [-verbose]
managesdk -listEnabledProfileAll [-verbose]
managesdk -enableProfile -profileName AppSrv01 -sdkname 1.7_64 -enableServers
managesdk -enableProfileAll -sdkname 1.7_64 -enableServers
managesdk -getNewProfileDefault [-verbose]
managesdk -setNewProfileDefault -sdkname 1.7_64
managesdk -getCommandDefault [-verbose]
managesdk -setCommandDefault -sdkname 1.7_64
```

### manageprofiles スクリプトを使用して WebSphere プロファイルを作成するには：

- 1 プロファイル作成対象のコンピューターで、コマンドプロンプトを開いて [appserver root]¥bin¥ ディレクトリに移動します。
- 2 次のコマンドを入力して WebSphere Profile Management Tool を起動します。
  - (Windows) manageprofiles.bat
  - (Linux/UNIX) ./manageprofiles.sh
- 3 適切なオプションを指定してコマンドを入力し、manageprofiles スクリプトでプロファイルを作成します。例えば、次のようにテキストを入力します。

- (ウィンドウ)：

```
[appserver root]\bin>manageprofiles.bat -create -templatePath
 "[appserver root]\profileTemplates\default" -profileName DS_AppSrv01
 -profilePath "[appserver root]\profiles\DS_AppSrv01" -isDefault
```

- (Linux または UNIX)：

```
[appserver root]/bin>./manageprofiles.sh -create -templatePath
 "[appserver root]/profileTemplates/default" -profileName DS_AppSrv01
 -profilePath "[appserver root]/profiles/DS_AppSrv01" -isDefault
```

### デプロイメント管理プロファイルの作成

```
[appserver root]\bin>manageprofiles.bat -create -profileName LC_Dmgr01 -profilePath "[appserver
 root]\profiles\LC_Dmgr01" -templatePath "[appserver root]\profileTemplates\management"
```

### 管理対象ノードプロファイルの作成

```
[appserver root]\bin>manageprofiles.bat -create -profileName LC_AppSrv01 -profilePath "[appserver
 root]\profiles\LC_AppSrv01" -templatePath "[appserver root]\profileTemplates\managed"
```

**注意：**上記のコマンドはプロファイル作成に必要な最小限のパラメータを使用しています。さらに次の引数を設定して、nodename、cellname および hostname を指定できます。

```
-nodeName
-cellName
-hostName
```

このコマンドで使用できるパラメーターについては、WebSphere Application Server ドキュメントの[この記事](#)を参照してください。

**注意：**コマンドラインで manageprofiles.sh help create と入力すると、manageprofiles スクリプトで指定可能なオプションのリストを表示できます。通常は、次のオプションを指定してください。

- プロファイルテンプレートが保存されているテンプレートパス。プロファイルパスは通常、[appserver root]/profileTemplates です。default、cell、dmgr、managed、management、および secureproxy などがプロファイルテンプレートの例となります。
- プロファイルのパス。プロファイルのパスについては、このドキュメントで使用する表記を参照してください。

- プロファイル名。プロファイル適用先の WebSphere Application Server を簡単に識別できるプロファイル名を指定します（例えば、AEM forms の場合は LC などの識別子を含めます）。
- プロファイルが WebSphere Application Server のデフォルトプロファイルの場合、-isDefault オプションを指定したプロファイルがデフォルトのプロファイルです。Network Deployment を使用するクラスターでは、dmgr01 プロファイルが通常はデフォルトのプロファイルとして存在します。

**注意：** WebSphere Application Server Environments 画面で選択するオプションによっては、プロファイルを作成するインストール後に、managedprofiles.bat または managedprofiles.sh を実行する必要があります。例えば、WebSphere Application Server Environments 画面で「None」を選択した場合、Deployment Manager 用に 1 回、アプリケーションサーバーノード用に 1 回、適切なコマンドを実行する必要があります。

### 1.3.2 グローバルセキュリティを有効にする場合の WebSphere Application Server の設定

グローバルセキュリティを使用する場合、適切なロールを持つユーザーとして WebSphere Application Server を実行する必要があります。WebSphere グローバルセキュリティを有効にする場合、次のいずれかの方法で WebSphere Application Server を設定できます。

- 必要なロールを持つ新規ユーザーを作成し、そのユーザーとして WebSphere Application Server を実行します。WebSphere Application Server を実行するユーザーが既に存在する場合は、必要なロールをそのユーザーに割り当てます。

**重要：** このユーザーとして WebSphere Application Server を起動します。グローバルセキュリティが有効になっているときに、他のユーザーとして WebSphere Application Server を起動すると、一部の WebSphere プロセスが停止する可能性があります。

安全性の高い環境では、この方法を使用することをお勧めします。

- 必要なロールを EVERYONE グループに設定します。

新しい WebSphere Application Server ユーザーを作成するには：

1 WebSphere Administrative Console のナビゲーションツリーで、**Environment / Naming / CORBA Naming Service Users** をクリックした後、右側のウィンドウの「Add」をクリックします。

2 「Roles」で、すべてのロールを選択します。

3 「Search and Select Users」で、「User Realm」を選択します。

4 検索ボックスに検索文字列を入力し、「Search」をクリックします。

**注意：** すべてのユーザーを取得するには、アスタリスク (\*) を入力します。

5 「Available」テキストボックスから、必要なユーザーを選択し、右矢印をクリックして「Mapped to role」ボックスに追加します。

6 「Save directly to master configuration」をクリックします。

既存の WebSphere Application Server ユーザーを設定するには：

1 WebSphere Administrative Console のナビゲーションツリーで、**Environment / Naming / CORBA Naming Service Users** をクリックした後、右側のウィンドウで該当ユーザーをクリックします。

2 「Roles」で、必要なロールを選択します。

3 「OK」または「Apply」をクリックします。

4 「Save directly to master configuration」をクリックします。

#### EVERYONE グループを設定するには

1 WebSphere Administrative Console のナビゲーションツリーで、**Environment** / **Naming** / **CORBA Naming Service Groups** をクリックします。

2 「**Roles**」で、必要なロールを選択します。

3 「**Select from special subjects**」を有効にし、Special subjects リストから「**EVERYONE**」グループを選択します。

**注意：**EVERYONE グループが設定済みの場合、グループは Special subjects リストに表示されません。このグループに必要なロールが割り当てられていない場合、割り当てする必要があります。

1 「**OK**」または「**Apply**」をクリックします。

2 「**Save directly to master configuration**」をクリックします。

### 1.3.3 WebSphere Application Server プロファイルの統合

ここで、WebSphere Application Server の各インスタンスに関して作成したプロファイルを Deployment Manager のプロファイルに追加することにより、個々のサーバーを統合する必要があります（6 ページの「[1.3.3.1 プロファイルの追加](#)」を参照）。

また、Deployment Manager のプロファイルからプロファイルを削除することにより、WebSphere Application Server インスタンスを既存の WebSphere セルから削除することもできます（7 ページの「[1.3.3.2 プロファイルの削除](#)」を参照）。

#### 1.3.3.1 プロファイルの追加

WebSphere Application Server のプロファイルを追加する前に、Deployment Manager が実行されていることを確認します。また、WebSphere Application Server インスタンスから Deployment Manager へ、Deployment Manager の名前および IP アドレスを使用して接続できることも確認します。

**重要：**WebSphere Application Server プロファイルを追加する前に、すべての WebSphere Application Server インスタンスのシステムクロックが同期していることを確認します。

#### カスタムプロファイルを Deployment Manager に追加するには：

1 Deployment Manager が実行されていない場合は、Deployment Manager プロファイルの bin ディレクトリに移動して、次のうち該当するスクリプトを実行します。

- (Windows) startManager.bat
- (Linux、UNIX) ./startManager.sh

「**Cell (deployment manager and a managed node)**」オプションを使用して WebSphere Application Server をインストールした場合は、[appserver root]/profiles/<profile\_name>/bin ディレクトリに移動し、次のうち該当するコマンドを実行してノードエージェントを起動します。

- (Windows) startNode.bat
- (Linux、UNIX) ./startNode.sh

**注意：**アプリケーションサーバーノードをセルに追加する場合を除き、アプリケーションサーバーノードプロファイルの startNode.bat または startNode.sh を実行する必要はありません。このノードをセルに追加したら、適切な startNode コマンドを実行してノードを起動できます。ノードをセルに追加する方法については、手順 3 を参照してください。

Deployment Manager と共にインストールした管理対象のノードについてのみ、startNode.bat または startNode.sh を実行します。

2 コマンドプロンプトから、追加する WebSphere Application Server インスタンスの [profiles root]/<profile name>/bin ディレクトリに移動します。

3 コンピューター名をパラメーターとして使用して addNode スクリプトを実行します。例えば、次のテキストを入力します。

- (Windows) addNode.bat [dmgr\_host] [dmgr\_port]
- (UNIX/Linux) ./addNode.sh [dmgr\_host] [dmgr\_port]

**注意：**[dmgr\_host] 引数は必須です。その他の引数はすべてオプションです。Deployment Manager のデフォルトの SOAP ポートの場合、デフォルトのポート番号は 8879 です。詳しくは、WebSphere Application Server ドキュメントの[この記事](#)を参照してください。

ノードをセルに統合するほかに、addNode では、ノードエージェントプロセスも開始されます。ノードがセルに統合された後に、ノードエージェントが startNode コマンドで開始されます。これも、WebSphere Application Server プロファイルの bin ディレクトリにあります。この処理中に、統合されるノードは、デフォルトでポート 8879 を使用して、Deployment Manager と通信します。

UNIX では、ノードエージェントをオペレーティングシステムのデーモンプロセスとして追加することをお勧めします。アプリケーションサーバーの基本インストールの bin ディレクトリにある WASService を使用すると、ノードエージェントを Windows のサービスとして追加できます。

### 1.3.3.2 プロファイルの削除

WebSphere Application Server プロファイルをセルから削除するには、該当するプロファイルを WebSphere Deployment Manager から削除します。このタスクは、スクリプトファイルのペアまたは Deployment Manager Administrative Console を使用して実行できます。

**注意：**プロファイルをセルから削除しても、サーバーがクラスターから削除されるだけで、プロファイルは削除されません。プロファイルは残り、必要に応じて後からセルに戻すこともできます。プロファイルを完全に削除するには、プロファイルを削除するための個別のタスクを行う必要があります。

不要になったプロファイルは、Deployment Manager およびサーバー上で削除できます。

**スクリプトファイルを使用して WebSphere Application Server を削除するには：**

1 Deployment Manager が実行されていない場合は、Deployment Manager プロファイルの bin ディレクトリに移動して、次のうち該当するスクリプトを実行します。

- (Windows) startManager.bat
- (Linux、UNIX) ./startManager.sh

2 削除対象の WebSphere Application Server ごとに、ノードエージェントを実行しているプロファイルの bin ディレクトリに移動して、次のうち該当する removeNode スクリプトを実行します。

- (Windows) removeNode.bat
- (Linux、UNIX) ./removeNode.sh

3 Deployment Manager プロファイルの bin ディレクトリに移動して、次のうち該当する cleanup スクリプトを実行します。

- (Windows) cleanupNode.bat <node name> [deploymgr host] [deploymgr port] [options]
- (Linux、UNIX) ./cleanupNode.sh <node name> [deploymgr host] [deploymgr port] [options]

詳しくは、WebSphere Application Server ドキュメントの[この記事](#)を参照してください。

**Deployment Manager を使用して WebSphere Application Server を削除するには：**

- 1 Deployment Manager が実行されていることを確認します。
- 2 Web ブラウザーで、Deployment Manager の URL を、http://<servername>:<port>/ibm/console のように入力します。

**注意：**デフォルトでは、Deployment Manager Administrative Console Web アプリケーションのリSPORTは 9060 です。

- 3 左側のウィンドウで、「**System Administration**」を選択して「**Nodes**」をクリックします。
- 4 削除するノードを選択して「**Remove Node**」をクリックします。
- 5 ノードが削除されたことを確認するには、**System Administration** / **Nodes** に移動して、ノードがリストに表示されないことを確認します。

**プロファイルを削除するには：**

- 1 コマンドプロンプトを開いて、[appserver root]/bin ディレクトリに移動します。
- 2 コンソールから次のコマンドを実行します。

- (Windows) manageprofiles.bat -delete -profileName [profileName]
- (UNIX または Linux) ./manageprofiles.sh -delete -profileName[ プロファイル名 ]

**注意：**プロファイルディレクトリとログファイルは削除されません。プロファイルディレクトリは手動で削除する必要があります。ディレクトリを削除せずに、削除したプロファイルと同じ名前を使用してプロファイルを作成しようとすると、エラーが発生します。

### 1.3.4 WebSphere クラスターの作成

ここで、次のタスクを実行して WebSphere クラスターを作成する必要があります。

- Deployment Manager を使用してクラスターを作成します。
- クラスターの分散環境を設定します。
- クラスターの WebSphere Application Server インスタンスのポートおよびホストエイリアスを追加します。

デフォルトでは、WebSphere Administrative Console Web アプリケーションのリSPORTは 9060 です。

**Deployment Manager を使用してクラスターを作成するには：**

- 1 Deployment Manager のホストコンピューターの WebSphere Administrative Console にログインします。
- 2 Web ブラウザーで、Deployment Manager の URL を、http://<servername>:<port>/ibm/console のように入力します。

**注意：**デフォルトでは、Deployment Manager Administrative Console Web アプリケーションのリSPORTは 9060 です。

- 3 WebSphere Administrative Console ナビゲーションツリーで、**Servers** / **Clusters** / **WebSphere application server clusters** をクリックし、「**New**」をクリックします。
- 4 「**Enter Basic Cluster Information**」ボックスに、クラスターの名前を、ds\_cluster のように入力します。
- 5 右側のウィンドウの「**Member name**」ボックスに、メンバー名を入力します。これは、クラスター内の最初のメンバーの名前です。
- 6 **Select Node** リストで、このクラスターメンバーが存在するノードを選択します。
- 7 「**Create the member using an application server template**」を選択し、リストから「**default**」を選択して、「**Next**」をクリックします。
- 8 「**Member Name**」ボックスに、クラスターに追加する別のメンバーの名前を入力します。
- 9 **Select Node** リストで、このクラスターメンバーが存在するノードを選択します。
- 10 「**Generate unique HTTP ports**」を選択して「**Add Member**」をクリックします。
- 11 手順 8 ~ 11 を繰り返し、新しいメンバー名を入力して、WebSphere Application Server をクラスターに追加します。

12 すべてのメンバーを追加したら、「Next」をクリックします。

13 「Finish」をクリックし、「Save」をクリックします。

14 **System Administration** / **Save Changes to Master Repository** をクリックして、「Synchronize changes with Nodes」を選択し、「Save」をクリックします。

クラスターの分散環境を設定するには：

1 WebSphere Administrative Console のナビゲーションツリーで、**System administration** / **Nodes** をクリックし、ノードがリストに表示されていること、ノードのステータスが「Synchronized」であること、クラスターのすべての WebSphere Application Server インスタンスが起動されていることを確認します。

2 **Servers** / **Server Types** / **WebSphere application servers** をクリックし、右側のウィンドウで、サーバー名をクリックします

3 「Configuration」タブをクリックし、「Container Setting」で、「Session management」をクリックします。

4 「Additional Properties」で、「Distributed environment settings」をクリックします。

5 「General Properties」の「None」をクリックし、「OK」をクリックします。

6 「Save directly to the master configuration」をクリックします。

7 次の画面の「Additional Properties」で、「Distributed Environment Settings」および「custom tuning parameters」をクリックします。

8 「Low (optimize for failover)」を選択して「OK」をクリックします。

9 ナビゲーションツリーで **Servers** / **Application servers** をクリックし、右側のウィンドウでサーバー名をクリックします。

10 「Performance」で「Performance Monitoring Infrastructure (PMI)」を選択します。

11 次の画面で、「Enable Performance Monitoring Infrastructure (PMI)」を選択します。

12 「Currently Monitored Statistics Set」で、「Basic」を選択して「OK」をクリックします。

13 クラスター内のサーバーごとに、手順 2 ~ 13 を繰り返します。

14 「Messages」ボックスで、「Save directly to the master configuration」をクリックします。

WebSphere Application Server ポートとエイリアスを設定するには：

1 Web ブラウザーで、Deployment Manager の URL を、`http://<servername>:<port>/ibm/console` のように入力します。

注意：デフォルトでは、WebSphere Administrative Console のリスポートは 9060 です。

2 ナビゲーションツリーで **Servers** / **Application servers** をクリックし、右側のウィンドウでサーバー名をクリックします。

3 次の画面の「Communications」で、「Ports」をクリックします。

4 表で「WC\_defaulthost」をクリックし、ポートアドレスを割り当てます。

注意：水平クラスターの場合、各サーバーは一意のまたは同じアドレスを持ちます。

5 クラスター内のサーバーごとに、手順 2 ~ 4 を繰り返します。

6 「Save directly to the master configuration」をクリックします。

7 ナビゲーションツリーで、**Environment** / **Virtual Hosts** をクリックし、右側のウィンドウで「default\_host」をクリックします。

8 「Additional Properties」で、「Host Aliases」をクリックします。

9 次の画面で、「New」をクリックして、クラスター内のサーバーに割り当てたポートを追加します。

- 10 「Host Name」ボックスに、アスタリスク (\*) を入力します。
- 11 手順 4 で割り当てた各ポートについて、手順 9 と 10 を繰り返します。
- 12 「OK」をクリックし、次に「Save directly to master configuration」をクリックします。
- 13 サーバーを再起動します。**Servers / Server Types / WebSphere application servers** をクリックし、サーバー名の横のチェックボックスを選択し、「Restart」をクリックします。

### 1.3.5 SOAP 接続タイムアウト設定の変更

クラスター内の各 WebSphere Application Server および Deployment Manager の SOAP 接続タイムアウトの設定を変更します。

SOAP 接続タイムアウト設定を変更するには

- 1 WebSphere Administrative Console にログインし、ナビゲーションツリーで、**Servers / Cluster / WebSphere application server clusters** をクリックします。
- 2 右側のウィンドウで、すべてのクラスターを停止します。
- 3 [appserver root]\profiles\\properties に移動して、テキストエディターで soap.client.props ファイルを開きます。
- 4 com.ibm.SOAP.requestTimeout プロパティを 1800 に設定します。
- 5 編集したファイルを保存します。
- 6 クラスター内のアプリケーションサーバーとデプロイメントマネージャごとに手順 3 ~ 5 を繰り返します。
- 7 デプロイメントマネージャー、ノード管理およびクラスターを起動します。

### 1.3.6 既存のクラスターへの新しいノードの追加

次の手順を実行してクラスターに新しいノードを追加します。

- 1 WebSphere Network Deployment ソフトウェアをインストールします。詳しくは 2 ページの「[1.2 WebSphere Network Deployment ソフトウェアのインストール](#)」を参照してください。
- 2 WebSphere Network Deployment ソフトウェアのインストール時にアプリケーションサーバーオプションを選択しなかった場合は、ここで WebSphere プロファイルを作成します。詳しくは、3 ページの「[1.3.1 WebSphere プロファイルの作成](#)」を参照してください。
- 3 Fix Pack および Feature Pack をインストールします。「[サポートされているプラットフォームの組み合わせ](#)」を参照してください。
- 4 タイムアウトを設定します。10 ページの「[1.3.5 SOAP 接続タイムアウト設定の変更](#)」を参照してください。
- 5 作成したプロファイルを統合します。詳しくは、6 ページの「[1.3.3 WebSphere Application Server プロファイルの統合](#)」を参照してください。
- 6 クラスターに新しいノードを追加します。
  - **Servers / Clusters / WebSphere application server clusters** をクリックします。
  - 右側のウィンドウで、ノードを追加するクラスターの名前をクリックします。
  - 右側のウィンドウの **Additional Properties / Cluster members** をクリックします。
  - 「New」をクリックします。
  - 「Create additional cluster members」画面で、新しいメンバーの名前を指定し、クラスターに追加するノードを選択します。

- 「**Add Member**」をクリックし、「**Next**」をクリックします。
  - 概要画面を確認して「**Finish**」をクリックします。
  - 「**Save directly to the master configuration**」をクリックします。
- 7 新しいノードを開始する前に、次のことを確認します。
- 必要なすべてのソフトウェアがインストールされ、環境変数が作成されている。
  - 一時ディレクトリの場所が新しいノードで使用できる。
  - GDS（共有）の場所が新しいノードで使用できる。
  - Adobe フォント、カスタマーフォントおよびシステムフォントのディレクトリが新しいノードで使用できる。
  - PDFG 設定が完了している。詳しくは、PDF Generator の設定を参照してください。
  - カスタムプロパティ、JVM 引数およびヒープ引数が新しいノード用に設定されている。既存のノードからこれらの設定をコピーできる場合があります。
  - 新しいノードで、既存のノードと同じ場所で Database jar ファイルを使用できる。新しいノード用にデータソースを作成しないでください。既に新しいノードで使用できます。
- 8 新しいノードを開始します。
- 注意：**新しいノードで、既存のノードと同じ場所ですべてのディレクトリ（ローカルおよび共有）を使用できることを確認してください。

## 1.4 WebSphere Application Server クラスターのテスト

WebSphere Application Server クラスターをテストして、すべてのメンバーがアクティブであること、クラスターが設計に従って動作することを確認できます。AEM forms のインストールおよび設定に進む前に、WebSphere Application Server クラスターが正常に機能することを確認する必要があります。

WebSphere Application Server クラスターをテストするには：

- 1 クラスターのすべての WebSphere Application Server インスタンスが起動していることを確認します。
- 2 [appserver root]/profiles/[profile name]/logs/[application server name]/SystemOut.log にある server.log ファイルを表示します。次のようなメッセージで、クラスターのアクティブなメンバーを確認できます。

```
[1/22/08 13:50:09:643 PDT] 00000018 PtpConnectedC I DCSV1031I: DCS Stack
DefaultCoreGroup.lc9_cluster at Member LCcell\Node01\Node01Server1:
Received a connection from an undefined member LCcell\Node02\
Node02Server1. Source address is /11.11.11.11.
[1/22/08 13:50:09:696 PDT] 0000001f RoleMember I DCSV8051I: DCS Stack
DefaultCoreGroup.lc9_cluster at Member LCcell\Node01\Node01Server1: Core
group membership set changed. Added: [LCcell\Node02\Node02Server1].
[1/22/08 13:50:09:704 PDT] 0000001d RecoveryDirec I CWRLS0012I: All
persistent services have been directed to perform recovery processing for
this WebSphere server (LCcell\Node01\Node01Server1).
[1/22/08 13:50:09:712 PDT] 00000018 MbuRmmAdapter I DCSV1032I: DCS Stack
DefaultCoreGroup.lc9_cluster at Member LCcell\Node01\Node01Server1:
Connected a defined member LCcell\Node02\Node02Server1.
[1/22/08 13:50:09:839 PDT] 00000020 RecoveryManag A WTRN0028I:
Transaction service recovering 0 transactions.
[1/22/08 13:50:26:744 PDT] 0000001f RoleMergeLead I DCSV8054I: DCS Stack
DefaultCoreGroup.lc9_cluster at Member LCcell\Node01\Node01Server1: View
change in process.
[1/22/08 13:50:26:764 PDT] 00000018 VSyncAlgo1 I DCSV2004I: DCS Stack
DefaultCoreGroup.lc9_cluster at Member LCcell\Node01\Node01Server1: View
synchronization completed successfully. The View Identifier is
(1:0.LCcell\Node01\Node01Server1). The internal details are None.
```

## 1.5 次の手順

次に、AEM forms ソリューションコンポーネントファイルをインストールする必要があります（「AEM forms モジュールのインストール」を参照）。

## 第 2 章：AEM forms モジュールのインストール

## 第 3 章: AEM Forms をデプロイするための設定

## 第4章：WebSphere クラスターの手動設定

この章では、クラスター環境で AEM forms の手動デプロイに備えて WebSphere Application Server クラスターを手動で設定する方法について説明します。この章の内容は、WebSphere Application Server クラスターを自動設定しないように選択している場合にのみ適用されます。アプリケーションサーバーを自動的に設定する方法について詳しくは、「デプロイのための AEM forms の設定」を参照してください。

インストールプロセスのこの段階では、既にファイルをインストールし、AEM forms Configuration Manager を実行してデプロイ可能な AEM forms アーカイブを設定しています。ここで、以下のタスクを手動で実行する必要があります。

- WebSphere Application Server を設定します（「16 ページの「[4.2 WebSphere Application Server インスタンスの設定](#)」を参照）。
- JDBS 接続を設定します（「19 ページの「[4.3 AEM forms データベース接続の設定](#)」を参照）。

### 4.1 ディレクトリ権限

AEM forms アプリケーションは、`[appserver root]/installedApps` ディレクトリにファイルを抽出します。これにより、そのディレクトリには書き込み権限を付与することが重要です。次の節では、書き込み権限を付与できない場合にファイルの抽出先を変更する方法について説明します。

**注意：** 抽出したファイルの位置を変更することをお勧めします。

#### 4.1.1 ファイルの抽出先の変更

- 1 WebSphere Administrative Console にログインします。
- 2 次のいずれかの操作を行います。
  - **Servers / Server Types / WebSphere Application servers** をクリックし、サーバー名（「`server1`」など）をクリックします。
- 3 「Server Infrastructure」で、**Java and forms workflow / Process Definition** をクリックします。
- 4 「Additional Properties」で「**Java Virtual Machine**」をクリックし、次に「**Custom Properties**」をクリックします。
- 5 「**New**」をクリックし、**adobeidp.RootDirectory** という名前のカスタムプロパティを作成します。
- 6 `adobeidp.RootDirectory` の値を、アドビのネイティブファイルの抽出先となるパス（例えば、`[appserver root]/profiles/<profile_name>/installedApps`）に設定します。
- 7 「**OK**」または「**Apply**」をクリックします。
- 8 「Messages」ボックスで「**Save directly to master configuration**」をクリックし、アプリケーションサーバーを再起動します。

## 4.2 WebSphere Application Server インスタンスの設定

次のタスクを行って、クラスターにインストールした WebSphere Application Server インスタンスを設定する必要があります。

- WebSphere のタイムアウト設定を変更します（「16 ページの「[4.2.1 WebSphere のタイムアウト設定の変更](#)」を参照）。
- JVM のプロパティを変更します（「16 ページの「[4.2.2 JVM のプロパティの変更](#)」を参照）。
- データベース用の J2C 認証エイリアスを作成します（「19 ページの「[4.2.3 データベース用の J2C 認証エイリアスの作成](#)」を参照）。

### 4.2.1 WebSphere のタイムアウト設定の変更

クラスター内の WebSphere Application Server ごとに、WebSphere のタイムアウト設定を変更する必要があります。

WebSphere のタイムアウト設定を変更するには：

- 1 WebSphere Administrative Console のナビゲーションツリーで、**Servers / Application servers** をクリックし、右側のウィンドウでサーバー名をクリックします。
- 2 「Container Settings」で、**Container services / Transaction Service** をクリックします。
- 3 「**Total transaction lifetime timeout**」ボックスに、600 と入力して「OK」をクリックします。
- 4 「Container Settings」で、**Container Services / ORB Service** をクリックします。
- 5 「**Request timeout**」ボックスに 360 と入力し、「**Locate Request Timeout**」ボックスに 600 と入力して、「OK」をクリックします。
- 6 「Server Infrastructure」で、**Administration / Administration Services** をクリックします。
- 7 次の画面で、「**JMX Connectors**」をクリックし、表内で「**SOAPConnector**」をクリックします。
- 8 次の画面で、「**Custom properties**」をクリックし、表内で「**requestTimeout**」をクリックします。
- 9 「Value」ボックスに、1800 と入力します。
- 10 「OK」をクリックし、次に「**Save directly to master configuration**」をクリックします。

### 4.2.2 JVM のプロパティの変更

AEM forms オプションを追加するには、AEM forms クラスター内の WebSphere Application Server インスタンスごとに、Java 仮想マシン（JVM）のプロパティを変更する必要があります。

**注意：**JVM パラメーターを変更した後、アプリケーションサーバーの各ノードを再起動する必要があります。

この手順を開始する前に、クラスターで 32 ビットまたは 64 ビット JVM が使用されているかどうかを確認する必要があります。クラスター設定に必要な JVM を判別するには、『AEM forms のインストールの準備（サーバークラスター）』を参照してください。

クラスターキャッシュの JVM 引数を正しく設定できるよう、この手順を開始する前に、AEM forms クラスターにクラスターキャッシュをどのように実装するか決定してください。クラスターキャッシュは、UDP または TCP を使用して実装できますが、両方使用することはできません。選択に影響する要因は次のとおりです。

- クラスターが IPv4 ベースの場合のみ、UDP を使用できます。
- クラスターが IPv4 ベースまたは IPv6 ベースの場合は、TCP を使用します。IPv6 ベースのクラスターには、IPv6 に準拠するように TCP を使用する必要があります。

また、TCP を使用してクラスターキャッシュを実装する場合、TCP ロケーターを正しく設定する必要があります  
（「キャッシュロケーターの設定（TCP のみを使用したキャッシュ）」を参照）。

 TCP プロトコルには継承の信頼性があるので、実稼働システムには UDP マルチキャストではなく TCP を使用することをお勧めします。

#### JVM のプロパティを変更するには：

1 WebSphere Administrative Console にログインし、ナビゲーションツリーで **Servers / Application servers** をクリックし、右側のウィンドウでサーバー名をクリックします。

2 「Server Infrastructure」で、**Java and forms workflow / Process Definition** をクリックします。

3 「Additional Properties」で、「**Java Virtual Machine**」をクリックして、次のプロパティを追加または設定します。

- 「**Initial Heap Size**」ボックスに 512 と入力します。
- 「**Maximum Heap Size**」ボックスに、以下のいずれかの値を設定します。
  - (32 ビット JVM のみ) 1024 と入力します。
  - (64 ビット JVM のみ) 4096 と入力します。
- 「**Generic JVM arguments**」ボックスに、以下の引数を追加します。

`-Xgcpolicy:gencon`

`-Dfile.encoding=utf8`

`-Dcom.adobe.livecycle.crx.integration.url=http://[IP]:[ポート]`

**注意：**-Xgcpolicy:gencon JVM 引数は、WebSphere で IBM JDK を使用する場合にのみ追加します。ただし、Solaris オペレーティングシステム上の WebSphere の場合は、この引数を追加しないでください。

- 「**Generic JVM arguments**」ボックスに、以下のいずれかの値を設定します。
  - (32 ビット JVM のみ) `-XX:MaxPermSize=256m` と入力します。
  - (64 ビット JVM のみ) `-XX:MaxPermSize=512m` と入力します。

4 同じ画面の「**Generic JVM arguments**」ボックスで、設定したクラスターキャッシュメカニズム（UDP または TCP）に応じて、次のキャッシュ引数を追加します。

#### • UDP 検索を使用したキャッシュ

- マルチキャストポート引数を次の形式で設定します。

`-Dadobe.cache.multicast-port=<port number>`

**注意：**<port number> の値は、1025 ~ 65535 の使用可能なポートにすることができます。マルチキャストポートは AEM forms クラスターに対して一意である必要があります（同じネットワーク上の別のクラスターが、そのポートを使用していないけません。同じネットワーク上の別のクラスターで同じポートを使用すると、ブートストラップに失敗します）。次の例のように、AEM forms クラスターのすべてのノードに同じ <port number> を設定することをお勧めします。

`-Dadobe.cache.multicast-port=33456`

- マルチキャストアドレス引数の設定はオプションです。IPv4 および IPv6 用のデフォルトのマルチキャストアドレスは、次のとおりです。

IPv6 - `FF38::1234`

IPv4 - `239.192.81.1`

ネットワークのマルチキャストアドレスに制限を設けている場合は、次の引数を使用してマルチキャストアドレスを設定します。

`-Dadobe.cache.multicast-address=<ip address>`

**注意:** <ip address> の値は、マルチキャストネットワークングに使用する IP アドレスです。adobe.cache.multicast-port がゼロの場合、IP アドレスは無視されます。

**注意:** マルチキャストアドレスは AEM forms クラスターに対して一意である必要があり、同じネットワーク上の別のクラスターがそのアドレスを使用してはいけません。AEM forms クラスターのすべてのノードに同じ <ip address> を設定することをお勧めします。以下に例を挙げます。

```
-Dadobe.cache.multicast-address=239.192.81.1
```

#### • TCP のみを使用したキャッシュ

- IPv4 の場合、クラスターロケータ引数を次の形式で設定します。

```
-Dadobe.cache.cluster-locators=<IPaddress>[<port number>],<IPaddress>[<port number>]
```

IPv6 の場合、クラスターロケータ引数を次の形式で設定します。

```
-Dadobe.cache.cluster-locators=<hostname>@<IPv6 address>[<port number>], <hostname>@<IPv6 address>[<port number>]
```

**注意:** クラスターのすべてのノードのロケータを、カンマ区切りのリストで設定します。<IPaddress> の値は、ロケータを実行するコンピューターの IP アドレスです。また、<port number> の値は、1025 ~ 65535 の未使用のポートです。次の例のように、すべてのロケータに同じ <port number> を設定することをお勧めします。

```
-Dadobe.cache.cluster-locators=10.20.30.5[22345],10.20.30.6[22345]
```

**注意:** すべてのノードに TCP ロケータを実行しないでください。2 つの TCP ロケータだけを設定します。1 つの TCP ロケータがプライマリロケータとして機能できるようにし、別の TCP ロケータがセカンダリ / バックアップロケータとして機能するようにします。TCP ロケータの設定の詳細については、「クラスターでのキャッシュロケータの設定 (TCP のみを使用したキャッシュ)」を参照してください。

- 複数のネットワークインターフェイスを持つマシンの場合

複数のネットワークインターフェイスカード (NIC) を介して複数のネットワークに接続するマシンもあります。そのようなマシンでは、JVM プロパティ -Dadobe.cache.bind-address を、forms Server に使用するネットワークインターフェイスカードの IP アドレスに設定します。

```
-Dadobe.cache.bind-address=<IP Address>
```

**注意:** 1 つのネットワークインターフェイスカードを持つマシンに対しても、JVM プロパティ -Dadobe.cache.bind-address を設定することをお勧めします。

- 5 アプリケーションサーバーへのサービス拒否攻撃を防ぐには、次の JVM 引数を設定します。

```
-DentityExpansionLimit=10000
```

- 6 「Apply」をクリックして「Custom Properties」をクリックします。

- 7 (IPv4 のみ) 次の画面で、「New」をクリックして次のプロパティを追加または設定し、「OK」をクリックします。

- 「Name」ボックスに java.net.preferIPv4Stack と入力します。
- 「Value」ボックスに true と入力します。

- 8 (IPv6 のみ) 次の画面で、「New」をクリックして次のプロパティを追加または設定し、「OK」をクリックします。

- 「Name」ボックスに java.net.preferIPv6Stack と入力します。
- 「Value」ボックスに true と入力します。
- 「Name」ボックスに java.net.preferIPv6Addresses と入力します。
- 「Value」ボックスに true と入力します。

- 9 「OK」をクリックし、次に「Save directly to master configuration」をクリックします。

- 10 サーバーを再起動します。

11 クラスター内のサーバーごとに、手順 11 ~ 19 を繰り返します。

### 4.2.3 データベース用の J2C 認証エイリアスの作成

データベース用の J2C 認証エイリアスを作成する必要があります。

データソース用の J2C 認証の設定を作成するには：

- 1 WebSphere Administrative Console のナビゲーションツリーで、**Security / Global security** をクリックします。
- 2 右側のウィンドウの「Authentication」で、**Java Authentication and Authorization Service / J2C authentication data** をクリックして、「New」をクリックします。
- 3 次のプロパティを設定します。
  - 「Alias」ボックスに、データベースユーザーに適切なエイリアス名（IDP\_DS/db2-db2user など）を入力します。
  - 「User ID」ボックスに、db2user のような名前を入力します。この ID は、IDP\_DS データソースで使用するデータベースにアクセスするためのログイン資格情報です。
  - 「Password」ボックスに、このユーザーのパスワードを入力します。

**注意：**このガイドでは、IDP\_DS は AEM forms データソースを識別します。

- 4 「OK」をクリックし、次に「Save directly to master configuration」をクリックします。
- 5 RM\_DS の場合、手順 3 と 4 を繰り返します。エイリアスとして EDC\_DS/db2-db2user を使用します。

**注意：**EDC\_DS は RM\_DS データソースの JNDI 名です。

## 4.3 AEM forms データベース接続の設定

WebSphere および AEM forms デプロイメントから AEM forms データベースに接続できるようにするには、データベースドライバーをインストールし、データソースを設定して、AEM forms のデータベース接続を作成する必要があります。

AEM forms データベースに使用するデータベースの種類に対応したドライバーをインストールする必要があります。ドライバーはアプリケーションサーバーのインストールディレクトリに配置する必要があります。

データベースに接続するデータソースを設定する必要があります。WebSphere では、DB2、Oracle、SQL Server のデータソースを設定できます。

『AEM forms のインストールの準備（サーバークラスター）』で行ったタスクに関する以下の情報が必要になります。

- データベース名
- サーバー名
- ポート番号
- ユーザー名
- パスワード

使用しているデータベースに応じて、次のいずれかの節を参照してください。

- 20 ページの「[4.3.1 DB2 データソースの設定](#)」
- 23 ページの「[4.3.2 Oracle データソースの設定](#)」
- 25 ページの「[4.3.3 SQL Server データソースの設定](#)」

### 4.3.1 DB2 データソースの設定

DB2 データソースを設定するには、まず DB2 データベースドライバをインストールする必要があります。次に、WebSphere 上に DB2 JDBC プロバイダーを作成してから、WebSphere 上にデータソースを作成し、対応する接続プールを設定する必要があります。

DB2 データベースドライバをインストールするには：

- 1 WebSphere Application Server インスタンスの [appserver root] ディレクトリで、db2libs という名前のディレクトリを作成します。
- 2 次のいずれかの場所から db2jcc.jar を [appserver root]\db2libs ディレクトリにコピーします。
  - [dbserver root] ディレクトリの下に java ディレクトリ ([dbserver root]\ibm\SqlLib\java (Windows)、[dbserver root]\java (Linux または UNIX) など)
  - [aem\_forms root]\lib\db\db2\
- 3 クラスター内の WebSphere Application Server ごとに、手順 1～2 を繰り返します。

DB2 JDBC プロバイダーを作成するには：

- 1 WebSphere Application Server インスタンスの、WebSphere Administrative Console のナビゲーションツリーで、**Environment / WebSphere Variables** をクリックし、右側のウィンドウで「**DB2UNIVERSAL\_JDBC\_DRIVER\_PATH**」をクリックします。
- 2 次の画面の「**Value**」ボックスに、db2libs ディレクトリのパスを入力します。
- 3 Cell Manager の範囲と同様に、各ノードの範囲について手順 1～2 を繰り返し、関連ノードの db2libs ディレクトリにパスを挿入します。
- 4 「**OK**」または「**Apply**」をクリックし、「**Messages**」ボックスの「**Save directly to master configuration**」をクリックします。
- 5 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックします。
- 6 表の上のドロップダウンリストで、「**Cluster=<cluster name>**」を範囲として選択し、「**New**」をクリックします。
- 7 Step 1 ウィンドウで、次のように設定し、「**Next**」をクリックします。
  - **Database type** リストで「**DB2**」を選択します。
  - **Provider type** リストで「**DB2 Universal JDBC Driver Provider**」を選択します。
  - **Implementation type** リストで「**Connection pool data source**」を選択します。各 Configuration Manager 設定スクリプトでは、フィールド実装クラス名は com.ibm.db2.jcc.DB2ConnectionPoolDataSource です。
  - 「**Name**」ボックスで、「**DB2 Universal JDBC Driver Provider**」の名前をそのまま残します。
- 8 Step 2 ウィンドウで、db2libs ディレクトリのパス ([appserver root]\db2libs など) を入力し、「**Next**」をクリックします。
- 9 Step 3 ウィンドウで、「**Finish**」をクリックして、「**Save directly to master configuration**」をクリックします。

DB2 JDBC データソースを作成するには：

- 1 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックし、右側のウィンドウで、プロバイダーをクリックします。
- 2 「**Additional Properties**」で「**Data sources**」をクリックし、「**New**」をクリックします。
- 3 Step 1 ウィンドウで値を次のように設定し、「**Next**」をクリックします。
  - 「**Data source name**」ボックスに Livecycle - DB2 - IDP\_DS と入力します。
  - 「**JNDI name**」ボックスに IDP\_DS と入力します。

4 Step 2 ウィンドウで、データベース名とサーバー名を入力します。

**注意：**データベースで使用されているポートがデフォルトのポート（50000）でない場合は、「**Port number**」ボックスで代替のポートも指定します。

5 「**Use this data source in container managed persistence (CMP)**」が選択されていることを確認します。

6 「**Next**」をクリックし、Step 3 ウィンドウで次の設定を指定します。

- 「**Component-managed authentication alias**」のリストで、「19 ページの「[4.2.3 データベース用の J2C 認証エイリアスの作成](#)」」でこのデータソース用に作成した認証エイリアスを選択します。
- **Mapping-configuration alias** リストで、「**DefaultPrincipalMapping**」を選択します。
- **Container-managed authentication alias** リストで、「19 ページの「[4.2.3 データベース用の J2C 認証エイリアスの作成](#)」」でこのデータソース用に作成した認証エイリアスを選択します。

7 「**Next**」をクリックし、Step 4 ウィンドウで「**Finish**」をクリックします。

8 「**Save directly to the master configuration**」をクリックします。

#### Lifecycle - DB2 - IDP\_DS 接続プールの設定

1 ナビゲーションツリーで、**Resources** / **JDBC** / **JDBC Providers** をクリックし、右側のウィンドウで、DB2 Universal JDBC Driver Provider をクリックします。「DB2 JDBC プロバイダーを作成するには」を参照してください。

2 「**Additional Properties**」で「**Data sources**」をクリックし、右側のウィンドウで「**Lifecycle - DB2 - IDP\_DS**」をクリックします。

3 次の画面の「**Additional Properties**」で「**Connection Pool Properties**」をクリックし、プロパティを次のように設定します。

- 「**Maximum connections**」ボックスに 30 と入力します。
- 「**Minimum connections**」ボックスに 1 と入力します。

4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### DB2 用のカスタムプロパティの設定

1 ナビゲーションツリーで、**Resources** / **JDBC** / **Data sources** をクリックし、右側のウィンドウで、データソースをクリックします。「DB2 JDBC データソースを作成するには」を参照してください。

2 「**Additional Properties**」で「**Custom Properties**」をクリックし、「**New**」をクリックします。

3 「**Name**」ボックスに useRRASetEquals と入力し、「**Value**」ボックスに true と入力します。

4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Document Security 用の DB2 JDBC データソースの作成

**注意：**この手順は、Document Security がインストール済みの場合にのみ該当します。

1 ナビゲーションツリーで、**Resources** / **JDBC** / **JDBC Providers** をクリックし、「DB2 JDBC プロバイダーを作成するには」で作成したプロバイダーをクリックします。

2 「**Additional Properties**」で「**Data sources**」をクリックし、「**New**」をクリックします。

3 Step 1 ウィンドウで値を次のように設定し、「**Next**」をクリックします。

- 「**Data source name**」ボックスに Lifecycle - DB2 - RM\_DS と入力します。
- 「**JNDI name**」ボックスに EDC\_DS と入力します。

- 「Component-Managed Authentication and XA Recovery Authentication Alias」のリストで、「19 ページの「4.2.3 データベース用の J2C 認証エイリアスの作成」」でこのデータソース用に作成した認証エイリアスを選択します。
- 4 Step 2 ウィンドウで、作成したデータベースのデータベース名とサーバー名を入力します。  
**注意：**データベースで使用されているポートがデフォルトのポート（50000）でない場合は、「Port number」ボックスで代替のポートも指定します。
  - 5 「Next」をクリックし、Step 3 ウィンドウで「Finish」をクリックします。
  - 6 前の手順で追加パラメーターを変更するために作成したデータソースを選択し、値を次のように設定します。
    - **Container-managed authentication alias** リストで、「19 ページの「4.2.3 データベース用の J2C 認証エイリアスの作成」」でこのデータソース用に作成した認証エイリアスを選択します。
    - **Mapping-configuration alias** リストで「DefaultPrincipalMapping」を選択します。
  - 7 「OK」または「Apply」をクリックし、「Save directly to master configuration」をクリックします。

#### Document Security 用の LiveCycle - DB2 - RM\_DS 接続プールの設定：

**注意：**この節は、Document Security がインストール済みの場合にのみ該当します。

- 1 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックし、右側のウィンドウで、「DB2 JDBC プロバイダーを作成するには」で作成した **DB2 Universal JDBC Driver Provider** をクリックします。
- 2 「Additional Properties」で「Data sources」をクリックし、右側のウィンドウで「Livecycle - DB2 - RM\_DS」をクリックします。
- 3 次の画面の「Additional Properties」で「Connection Pool Properties」をクリックし、プロパティを次のように設定します。
  - 「Minimum connections」ボックスに 20 と入力します。
  - 「Minimum connections」ボックスに 5 と入力します。
- 4 「OK」または「Apply」をクリックし、「Save directly to master configuration」をクリックします。

#### DB2 用のカスタムプロパティの設定：

- 1 ナビゲーションツリーで、**Resources / JDBC / Data sources** をクリックし、右側のウィンドウで、「Document Security 用の DB2 JDBC データソースの作成」で作成したデータソースをクリックします。
- 2 「Additional Properties」で「Custom Properties」をクリックし、「New」をクリックします。
- 3 「Name」ボックスに useRRASetEquals と入力し、「Value」ボックスに true と入力します。
- 4 「OK」または「Apply」をクリックし、「Save directly to master configuration」をクリックします。

#### 4.3.1.1 デフォルトの分離レベルの設定

- 1 WebSphere Integrated Solutions Console にログインします。
- 2 WebSphere Administrative Console のナビゲーションツリーで、**Resources / JDBC / Data Sources** をクリックします。
- 3 右側のペインのドロップダウンリストから、クラスターに **Cluster=[appropriate cluster name]** を選択します。対応するクラスター下のすべてのデータソースが表示されます。
- 4 JNDI 名が **IDP\_DS** である「LiveCycle - DB2 - IDP\_DS」をクリックします。
- 5 「Custom Properties」をクリックします。
- 6 **webSphereDefaultIsolationLevel** プロパティを探し、クリックして、編集用を開きます。

- 7 値を **2** に設定します。値 **2** は Read Committed を意味します。
- 8 「**Apply**」をクリックし、「**OK**」をクリックします。
- 9 JNDI 名 **EDC\_DS** を持つ **LiveCycle - DB2 - RM\_DS** で手順 5 ～ 8 を繰り返します。
- 10 ページ上部の「**Messages**」ボックスで、「**Save directly to master configuration**」をクリックします。
- 11 WebSphere を再起動します。

## 4.3.2 Oracle データソースの設定

Oracle データソースを構成するには、まず Oracle データベースドライバをインストールする必要があります。次に、WebSphere 上に Oracle JDBC プロバイダーを作成してから、WebSphere 上にデータソースを作成し、対応する接続プールを構成する必要があります。

### Oracle データベースドライバのインストール

- 1 WebSphere Application Server インスタンスごとに、[appserver root] ディレクトリに db\_driver という名前のディレクトリを作成します。
- 2 [DVD\_root]/third\_party/db/oracle ディレクトリの JDK 1.6 ドライバファイル ojdbc6.jar を、手順 1 で作成したディレクトリにコピーします。

### Oracle JDBC プロバイダーの作成

- 1 WebSphere Application Server インスタンスの、WebSphere Administrative Console のナビゲーションツリーで、**Environment / WebSphere Variables** をクリックし、右側のウィンドウで「**ORACLE\_JDBC\_DRIVER\_PATH**」をクリックします。
- 2 「**General Properties**」の「**Value**」ボックスに、「Oracle データソースの設定」で作成した ojdbc6.jar ファイルのパスを入力し、「**OK**」をクリックします。
- 3 WebSphere Application Server インスタンスごとに、手順 1 ～ 2 を繰り返して、各 WebSphere Application Server インスタンスが存在するノードの db\_driver ディレクトリの適切なパスを挿入します。
- 4 「**Save directly to master configuration**」をクリックします。
- 5 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックします。
- 6 表の上のドロップダウンリストで、「Cluster=<cluster\_name>」を範囲として選択し、「**New**」をクリックします。
- 7 Step 1 ウィンドウで値を次のように設定し、「**Next**」をクリックします。
  - **Database type** リストで「**Oracle**」を選択します。
  - **Provider type** リストで「**Oracle JDBC Driver**」を選択します。
  - **Implementation type** リストで「**Connection pool data source**」を選択します。
- 8 Step 2 ウィンドウで、デフォルトのデータベースクラスパスをそのまま使用し、「**Next**」をクリックします。
- 9 Step 3 ウィンドウで、「**Finish**」をクリックして、「**Save directly to master configuration**」をクリックします。

### Oracle JDBC データソースの作成

- 1 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックし、右側のウィンドウで、「Oracle データソースの設定」で作成したプロバイダーをクリックします。
- 2 「**Additional Properties**」で「**Data sources**」をクリックし、「**New**」をクリックします。
- 3 Step 1 ウィンドウで値を次のように設定し、「**Next**」をクリックします。
  - 「**Data source name**」ボックスに Livecycle - oracle - IDP\_DS と入力します。

- 「**JNDI name**」ボックスに IDP\_DS と入力します。
  - 「**Component-Managed Authentication and XA Recovery Authentication**」のリストで、「19 ページの「[4.2.3 データベース用の J2C 認証エイリアスの作成](#)」」でこのデータソース用に作成した認証エイリアスを選択します。
- 4 Step 2 ウィンドウで、「**URL**」ボックスに次の行を入力し、「**Next**」をクリックします。
- ```
jdbc:oracle:thin:@<server_host>:<port>:<SID>
```
- <server\_host> はデータベースサーバーの IP アドレス、<port> はデータベースがリスンしているポート（デフォルトは 1521）、<SID> はデータベースのサービス ID です。
- 5 Step 3 ウィンドウで、「**Finish**」をクリックして、「**Save directly to master configuration**」をクリックします。
- 6 前の手順で追加パラメーターを変更するために作成したデータソースを選択し、オプションを次のように設定します。
- **Container-managed authentication alias** リストで、「19 ページの「[4.2.3 データベース用の J2C 認証エイリアスの作成](#)」」でこのデータソース用に作成した認証エイリアスを選択します。
  - **Mapping-configuration alias** リストで「**DefaultPrincipalMapping**」を選択します。
- 7 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Lifecycle - oracle - IDP\_DS 接続プールの設定

- 1 ナビゲーションツリーで、**Resources** / **JDBC** / **JDBC Providers** をクリックし、右側のウィンドウで、Oracle JDBC ドライバーデータソースをクリックします。「Oracle JDBC プロバイダーの作成」を参照してください。
- 2 「Additional Properties」で「**Data sources**」をクリックし、右側のウィンドウで「**Lifecycle - oracle - IDP\_DS**」をクリックします。
- 3 次の画面の「Additional Properties」で「**Connection Pool Properties**」をクリックし、「**Maximum connections**」ボックスに 30 と入力します。
- 4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Oracle 用のカスタムプロパティの設定

- 1 ナビゲーションツリーで、**Resources** / **JDBC** / **Data sources** をクリックし、右側のウィンドウで、「Oracle JDBC データソースの作成」で作成したデータソースをクリックします。
- 2 「Additional Properties」で「**Custom Properties**」をクリックし、「**New**」をクリックします。
- 3 「**Name**」ボックスに useRRASetEquals と入力し、「**Value**」ボックスに true と入力します。
- 4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Document Security 用の Oracle JDBC データソースの作成

**注意：**この節は、Document Security がインストール済みの場合にのみ該当します。

- 1 ナビゲーションツリーで、**Resources** / **JDBC** / **JDBC Providers** をクリックし、右側のウィンドウで、「Oracle JDBC プロバイダーの作成」で作成したプロバイダーをクリックします。
- 2 「Additional Properties」で「**Data sources**」をクリックし、「**New**」をクリックします。
- 3 Step 1 ウィンドウで値を次のように設定し、「**Next**」をクリックします。
  - 「**Data source name**」ボックスに Lifecycle - oracle - RM\_DS と入力します。
  - 「**JNDI name**」ボックスに EDC\_DS と入力します。
  - 「**Component-Managed Authentication and XA Recovery Authentication Alias**」のリストで、「19 ページの「[4.2.3 データベース用の J2C 認証エイリアスの作成](#)」」でこのデータソース用に作成した認証エイリアスを選択します。

- 4 Step 2 ウィンドウで、「URL」ボックスに次の行を入力し、「Next」をクリックします。

```
jdbc:oracle:thin:@<server_host>:<port>:<SID>
```

<server\_host> はデータベースサーバーの IP アドレス、<port> はデータベースがリスンしているポート（デフォルトは 1521）、<SID> はデータベースのサービス ID です。

- 5 Step 3 ウィンドウで、「Finish」をクリックして、「Save directly to master configuration」をクリックします。
- 6 前の手順で追加パラメーターを変更するために作成したデータソースを選択し、オプションを次のように設定します。
- **Container-managed authentication alias** リストで、「19 ページの「4.2.3 データベース用の J2C 認証エイリアスの作成」」でこのデータソース用に作成した認証エイリアスを選択します。
  - **Mapping-configuration alias** リストで「DefaultPrincipalMapping」を選択します。
- 7 「OK」をクリックし、次に「Save directly to master configuration」をクリックします。

#### Document Security 用の LiveCycle - oracle - RM\_DS 接続プールの設定：

**注意：**この節は、Document Security がインストール済みの場合にのみ該当します。

- 1 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックし、右側のウィンドウで、「Oracle JDBC プロバイダーの作成」で作成した Oracle JDBC ドライバープロバイダーをクリックします。
- 2 「Additional Properties」で「Data sources」をクリックし、右側のウィンドウで「Livecycle - oracle - RM\_DS」をクリックします。
- 3 次の画面の「Additional Properties」で「Connection Pool Properties」をクリックし、プロパティを次のように設定します。
  - 「Minimum connections」ボックスに 20 と入力します。
  - 「Minimum connections」ボックスに 1 と入力します。
- 4 「OK」または「Apply」をクリックし、「Save directly to master configuration」をクリックします。

#### Oracle 用のカスタムプロパティの設定：

- 1 ナビゲーションツリーで、**Resources / JDBC / Data sources** をクリックし、右側のウィンドウで、「Document Security 用の Oracle JDBC データソースの作成」で作成したデータソースをクリックします。
- 2 「Additional Properties」で「Custom Properties」をクリックし、「New」をクリックします。
- 3 「Name」ボックスに useRRASetEquals と入力し、「Value」ボックスに true と入力します。
- 4 「OK」または「Apply」をクリックし、「Save directly to master configuration」をクリックします。

### 4.3.3 SQL Server データソースの設定

SQL Server データソースを設定するには、まず SQL Server データベースドライバーをインストールする必要があります。次に、WebSphere 上に SQL Server JDBC プロバイダーを作成してから、WebSphere 上にデータソースを作成し、対応する接続プールを設定する必要があります。

#### SQL Server データベースドライバーのインストール

- (IBM JDK 6 の場合) ドライバーのインストールをまだ実行していない場合は、[aem-forms root]\lib\db\mssql\java6 ディレクトリから [appserver root]\lib\ ディレクトリに SQL Server JDBC Driver をコピーしてください。

(IBM JDK 7 の場合) ドライバーのインストールをまだ実行していない場合は、[aem-forms root]\lib\db\mssql ディレクトリから [appserver root]\lib\ ディレクトリに SQL Server JDBC Driver をコピーしてください。

**注意：**システム上でドライバーをインストールしたディレクトリの場所を控えておいてください。

### SQL Server JDBC プロバイダーの作成

- 1 WebSphere Administrative Console のナビゲーションツリーで、**Environment** / **Websphere Variables** をクリックし、右側のウィンドウで「MICROSOFT\_JDBC\_DRIVER\_PATH」をクリックします。..
- 2 「**General Properties**」の「**Value**」ボックスに、作成した sqljdbc.jar ファイルのパスを入力し、「**OK**」をクリックします。
- 3 「**Messages**」ボックスで、「**Save directly to master configuration**」をクリックします。
- 4 ナビゲーションツリーで、**Resources** / **JDBC** / **JDBC Providers** をクリックします。
- 5 右側のウィンドウの **Scope** ドロップダウンリストで、レベルとして「**Cluster=<cluster name>**」を選択し、「**New**」をクリックします。
- 6 **Create new JDBC provider** ウィンドウで、値を次のように設定し、「**Next**」をクリックします。
  - **Database type** リストで「SQL Server」を選択します。
  - **Provider type** リストで「Microsoft SQL Server JDBC Driver」を選択します。
  - **Implementation type** リストで「Connection Pool Data Source」を選択します。
  - 「**Name**」ボックスで、Microsoft SQL Server JDBC Driver と入力するか、デフォルト値をそのまま使用します。
- 7 **Enter database class path information** ウィンドウで、既存のエントリを次のいずれかに置き換えて、「**Next**」をクリックします。
  - `${MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc.jar`

注意：WebSphere 変数 MICROSOFT\_JDBC\_DRIVER\_PATH が設定されていると、データベースクラスパスの情報が自動的に設定されます。
- 8 **Summary** ウィンドウで、「**Finish**」をクリックして、「**Save directly to master configuration**」をクリックします。

### LiveCycle 用の SQL Server データソースの作成

次の手順に従って、使用しているアプリケーションサーバーのバージョンに適した SQL Server データソースを作成します。

- 1 ナビゲーションツリーで、**Resources** / **JDBC** / **JDBC Providers** をクリックし、右側のウィンドウで、「SQL Server JDBC プロバイダーの作成」で作成したプロバイダーをクリックします。
- 2 「**Additional Properties**」で「**Data sources**」をクリックし、「**New**」をクリックします。
- 3 **Enter basic data source information** ウィンドウで、値を次のように設定し、「**Next**」をクリックします。
  - 「**Data source name**」ボックスに Livecycle - SQLServer - IDP\_DS と入力します。
  - 「**JNDI name**」ボックスに IDP\_DS と入力します。
- 4 「**Enter database specific properties for the data source**」で、データベース名、サーバー名およびポートを入力します。
- 5 **Setup security aliases** ウィンドウで、次の設定を行って、「**Next**」をクリックします。
  - **Component managed authentication alias** リストで、「データソース用の J2C 認証の設定を作成するには」でこのデータソース用に作成した認証エイリアスを選択します。
  - **Mapping-configuration alias** リストで「**DefaultPrincipalMapping**」を選択します。
  - **Container managed authentication alias** リストで、「データソース用の J2C 認証の設定を作成するには」でこのデータソース用に作成した認証エイリアスを選択します。
- 6 **Summary** ウィンドウで、「**Finish**」をクリックして、「**Save directly to master configuration**」をクリックします。

7 データソースのデータストアヘルパクラスを設定します。次のタスクを実行します。

- ナビゲーションツリーで、**Resources / JDBC / Data sources** をクリックし、右側のウィンドウで、作成したデータソースをクリックします。
- 次の画面の「**Data store helper class name**」で、「**Specify a user-defined data store helper**」を選択し、既存のエントリを次のテキストに置き換えます。

```
com.ibm.websphere.rsadapter.GenericDataStoreHelper
```

8 ステートメントキャッシュサイズを変更します。次のタスクを実行します。

- WebSphere Administrative Console で、**JDBC / Data Sources** をクリックします。
- 作成したデータソースをクリックし、「**Additional Properties**」で「**WebSphere Application Server data source properties**」をクリックします。
- 「**Statement cache size**」フィールドの値を 80 に変更します。
- 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Lifecycle - SQLServer - IDP\_DS 接続プールの設定

1 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックし、右側のウィンドウで、既に作成したプロバイダーをクリックします

- **Microsoft SQL Server JDBC Driver**

2 「**Additional Properties**」で「**Data sources**」をクリックし、「**Lifecycle - SQLServer - IDP\_DS**」を選択します。

3 次の画面の「**Additional Properties**」で「**Connection Pool Properties**」をクリックし、「**Maximum connections**」ボックスに 30 と入力します。

4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### SQL Server 用のカスタムプロパティの設定

1 ナビゲーションツリーで、**Resources / JDBC / Data sources** をクリックし、右側のウィンドウで、「**LiveCycle 用の SQL Server データソースの作成**」セクションで作成したデータソースをクリックします。

2 「**Additional Properties**」で「**Custom properties**」をクリックし、「**New**」をクリックします。

3 「**Name**」ボックスに useRRASetEquals と入力し、「**Value**」ボックスに true と入力します。

4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Document Security 用の SQL Server データソースの作成

次の手順に従って、使用しているアプリケーションサーバーのバージョンに適した SQL Server データソースを作成します。

1 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックし、右側のウィンドウで、「**SQL Server JDBC プロバイダーの作成**」で作成したプロバイダーをクリックします。

2 「**Additional Properties**」で「**Data sources**」をクリックし、「**New**」をクリックします。

3 **Enter basic data source information** ウィンドウで、値を次のように設定し、「**Next**」をクリックします。

- 「**Data source name**」ボックスに Lifecycle - SQLServer - RM\_DS と入力します。
- 「**JNDI name**」ボックスに EDC\_DS と入力します。

4 **Enter database specific properties for the data source** ウィンドウで、「**Data store helper class name**」ボックスの既存のエントリを次の値に置き換えます。

```
com.ibm.websphere.rsadapter.GenericDataStoreHelper
```

5 **Setup security aliases** ウィンドウで、次の設定を行って、「**Next**」をクリックします。

- **Component managed authentication alias** リストで、「データソース用の J2C 認証の設定を作成するには」でこのデータソース用に作成した認証エイリアスを選択します。
- **Mapping-configuration alias** リストで「**DefaultPrincipalMapping**」を選択します。
- **Container managed authentication alias** リストで、「データソース用の J2C 認証の設定を作成するには」でこのデータソース用に作成した認証エイリアスを選択します。

6 **Summary** ウィンドウで、「**Finish**」をクリックして、「**Save directly to master configuration**」をクリックします。

7 ステートメントキャッシュサイズを変更します。次のタスクを実行します。

- WebSphere Administrative Console で、**JDBC / Data Sources** をクリックします。
- 作成したデータソースをクリックし、「**Additional Properties**」で「**WebSphere Application Server data source properties**」をクリックします。
- 「**Statement cache size**」フィールドの値を 80 に変更します。
- 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Lifecycle - SQLServer - RM\_DS 接続プールの設定

1 ナビゲーションツリーで、**Resources / JDBC / JDBC Providers** をクリックし、右側のウィンドウで、既に作成したプロバイダーをクリックします

- **SQL Server Provider**

2 「**Additional Properties**」で「**Data sources**」をクリックし、「**Lifecycle - SQLServer - RM\_DS**」を選択します。

3 次の画面の「**Additional Properties**」で「**Connection Pool Properties**」をクリックし、「**Maximum connections**」ボックスに 20 と入力します。

4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### SQL Server 用のカスタムプロパティの設定

1 ナビゲーションツリーで、**Resources / JDBC / Data sources** をクリックし、右側のウィンドウで、作成したデータソースをクリックします。

2 「**Additional Properties**」で「**Custom properties**」をクリックし、「**New**」をクリックします。

3 「**Name**」ボックスに useRRASetEquals と入力し、「**Value**」ボックスに true と入力します。

4 「**OK**」または「**Apply**」をクリックし、「**Save directly to master configuration**」をクリックします。

#### Windows 上での統合セキュリティの設定

1 ナビゲーションツリーで、**Resources / JDBC / Data Sources** をクリックし、右側のウィンドウで「**IDP\_DS**」をクリックします。

2 右側のウィンドウの「**Additional Properties**」で「**Custom Properties**」をクリックし、次の画面で「**integratedSecurity**」をクリックします。

3 次の画面の「**General Properties**」で、「**Value**」ボックスに true と入力します。

4 ナビゲーションツリーで、**Resources / JDBC / Data Sources** をクリックし、右側のウィンドウで「**Lifecycle - SQLServer - RM\_DS**」をクリックします。

5 右側のウィンドウの「**Additional Properties**」で「**Custom Properties**」をクリックし、次の画面で「**integratedSecurity**」をクリックします。

6 次の画面の「**General Properties**」で、「**Value**」ボックスに true と入力します。

- 7 「Apply」をクリックして、「Save directly to master configuration」をクリックします。
- 8 WebSphere がインストールされているコンピューター上で、Windows システムパス (C:\Windows) に sqljdbc\_auth.dll ファイルを追加します。sqljdbc\_auth.dll ファイルは、Microsoft SQL JDBC 4.0 ドライバーのインストール先ディレクトリ (デフォルトは [InstallDir]\sqljdbc\_3.0\enu\auth\x86) と同じ場所にあります。
- 9 次のタスクを実行して、WebSphere Application Server[ ノード名 ]を開始する Windows サービスの「ログオン」プロパティを変更します。
  - スタート／設定／コントロールパネル／管理ツール／サービスををクリックします。
  - 「[ ノード名 ]」を右クリックして、「プロパティ」を選択します。
  - 「ログオン」タブで「アカウント」を選択し、ローカルシステム以外の有効なユーザーアカウントを選択して、「OK」をクリックします。
- 10 SQL Server のセキュリティを「混合モード」から「Windows 認証のみ」に変更します。

## 4.4 次の手順

WebSphere Application Server クラスターを設定したら、以下のタスクを実行します。

- Configuration Manager を使用して、LiveCycle EAR ファイルを設定します（「デプロイのための LiveCycle の設定」を参照）。
- 以下のいずれかの方法を選択して、LiveCycle EAR ファイルを WebSphere Application Server クラスターにデプロイします。
  - 自動：Configuration Manager を使用します（「デプロイのための LiveCycle の設定」を参照）。
  - 手動：「付録 - WebSphere への手動デプロイ」を参照してください。

## 第5章：デプロイメント後のタスク

## 第6章：ロードバランシングの設定

ロードバランシング機能が動作するように WebSphere クラスターを設定できます。WebSphere Application Server に付属の IBM HTTP Server を使用して、以下のタスクを実行します。

- 準備作業を実行します (31 ページの「[6.1 インストールの準備](#)」)。
- IBM HTTP Server をインストールします (31 ページの「[6.2 Web サーバーのインストール](#)」)
- Web サーバープラグインをインストールします (32 ページの「[6.3 Web サーバープラグインのインストール](#)」)

**注意：**クラスター環境の LiveCycle でロードバランシング用にサポートされているのは、スティッキーセッションのみです。IBM HTTP サーバーでは、デフォルトでスティッキーセッションをサポートしています。

### 6.1 インストールの準備

Web サーバーをインストールする前に、次の設定タスクを実行します。

**サーバードメイン：**ドメインアーキテクチャを使用している場合は、該当するサーバーが、LDAP サーバーと同じドメインのメンバーであることを確認します。

**ローカルユーザーの作成：**Microsoft Windows で、IBM HTTP Server をサービスとして実行する場合は、ローカルアカウントを作成し、このアカウントをローカル管理者グループに含めることができます。

### 6.2 Web サーバーのインストール

次の手順では、WebSphere Network Deployment や WebSphere の基本的なアプリケーションサーバーとは別のノードに IBM HTTP Server をインストールする方法を説明します。Apache、Microsoft IIS、および Sun Java System Web Server など、IBM でサポートされているその他の Web サーバーのインストールおよび設定について詳しくは、IBM の Web ページ「[Web サーバー設定ファイルの編集](#)」を参照してください。

**注意：**WebSphere Application Server Network Deployment を持っていること、および補助インストールファイルがローカルで使用できるようになっていることを確認してください。

最初に、インストールメディアを挿入するか、ファイルをローカルディレクトリにコピーする必要があります。

**注意：**この手順は、既存のインストールを置き換えて IBM HTTP Server の現在のバージョンをアップグレードする場合にも使用できます。

- 1 インストールを開始するには、WebSphere Application Server Network Deployment インストーラーを含むディレクトリに移動して、次のうち該当するコマンドを入力します。
  - (Linux または UNIX) `./launchpad.sh`
  - (Windows) `launchpad.bat`
- 2 Launch Pad から、「**IBM HTTP Server 用のインストールウィザードを起動**」を選択し、補助の抽出インストールファイルの最初の部分の場所を指定します。
- 3 ようこそ画面で「**次へ**」をクリックします。
- 4 「**IBM の条件および IBM 以外の条件の両方に同意する**」を選択し、「**次へ**」をクリックします。
- 5 インストールディレクトリの場所を指定して、「**Next**」をクリックします。

- 6 HTTP ポートと HTTP Administration ポートを指定して、「Next」をクリックします。
- 7 (Windows)「Run IBM HTTP Server as a Windows Service」、「Run IBM HTTP Administration as a Windows Service」および「Log on as local system account」を選択します。  
**注意：**この選択には、ユーザー名とパスワードは不要です。特定のユーザーアカウントとパスワードの組み合わせを使用してこのサービスを実行するには、「Log on as a specified user account」を選択し、ユーザー ID とパスワードの情報を指定します。
- 8 「Startup Type」で、「Automatic」を選択し、「Next」をクリックします。
- 9 ユーザー ID とパスワードを指定して、HTTP 管理サーバーを作成し、「次へ」をクリックします。
- 10 「IBM WebSphere Application Server Web サーバー定義のために IBM HTTP Server プラグインをインストールする」オプションのチェックを解除し、「次へ」をクリックします。
- 11 インストール概要画面で選択内容を確認し、指定した内容を変更する場合は「戻る」をクリックします。「次へ」をクリックすると IBM HTTP Server のインストールを開始します。  
 インストールの状況が表示された後、ウィザードに、インストールの成功を示す完了ステータスパネルが表示されます。
- 12 「Finish」をクリックします。

## 6.3 Web サーバープラグインのインストール

アプリケーションサーバーがインストールされ、アプリケーションがデプロイされたら、Web サーバープラグインを HTTP サーバーにインストールします。この手順では、HTTP サーバーがクラスター以外のノードにあることを前提としています。

- 1 Web サーバーコンピューター（HTTP サーバーがインストールされているリモートシステム）で、WebSphere Network Deployment インストーラーディレクトリに移動し、該当するコマンドを入力して Launch Pad を実行します。
  - (Linux または UNIX) ./launchpad.sh
  - (Windows) launchpad.bat
- 2 Launch Pad から、「Launch the installation wizard for Web server plug-ins」を選択します。
- 3 「Installation roadmap」および「Plug-ins section of the Getting Started guide」の選択を解除して、「Next」をクリックします。
- 4 「IBM の条件および IBM 以外の条件の両方に同意する」を選択し、「次へ」をクリックします。インストーラーでシステムのチェックが行われます。
- 5 システムの前提条件確認画面で、「次へ」をクリックします。
- 6 システムが必要条件を満たしていることが確認されたら、「Next」をクリックします。  
**注意：**システムが必要条件を満たしていないことが確認された場合は、インストールを停止し、問題を修正してから、インストールを再開します。
- 7 「IBM HTTP Server V7」を選択して「Next」をクリックします。
- 8 「Web server machine (remote)」を選択して「Next」をクリックします。
- 9 [plugins\_root] ディレクトリと、Web サーバープラグインのインストール先を指定して、「Next」をクリックします。
- 10 「Select the existing IBM HTTP Server httpd.conf file」で、「Browse」をクリックし、[webserver root]/conf ディレクトリから httpd.conf ファイルを選択します。ここで、[webserver root] は、IBM HTTP Server のインストール先ディレクトリです。

- 11 「Specify the Web server port」ボックスで、デフォルトのポートである 80 をそのまま指定して、「Next」をクリックします。
- 12 「Specify a unique Web server definition name」ボックスに、この定義の固有な識別子を入力して、「Next」をクリックします。
- 13 「Web server plugin-cfg.xml file」ボックスで、デフォルトの設定をそのまま指定して、「Next」をクリックします。
- 14 「Host name or IP address for the Application Server」ボックスに、ND ノードのホスト名または IP アドレスを入力して、「Next」をクリックします。
- 15 確認ウィンドウで「Next」をクリックし、概要情報ウィンドウで「Next」をクリックします。
- 16 Web サーバープラグインがインストールおよびコピーされたら、「次へ」をクリックし、「完了」をクリックします。
- 17 IBM HTTP Server <plugin\_dir>/bin から適切なファイルを WebSphere Network Deployment [appserver root]\bin フォルダにコピーします。
  - (Windows) [webserver definition name].bat を設定します。例えば、configureserver1.bat
  - (Linux または UNIX) configure[webserver definition name].sh
- 18 Deployment Manager が実行されていることを確認し、Windows の場合は `configure<webserver definition name>.bat`、Linux または UNIX の場合は `configure<webserver definition name>.sh` を WebSphere Network Deployment コンピューターで実行して、WebSphere Network Deployment コンピューター上に管理対象外のノードを作成し、そこに Web サーバーを追加します。
- 19 WebSphere Administrative Console にログインし、ナビゲーションツリーで **Servers / Web servers** をクリックし、Web サーバー名をクリックします。「追加プロパティ」で、「リモート Web サーバー管理」をクリックします。ポート、ユーザー名、パスワードの詳細が、IBM HTTP 管理サーバーのために提供されているものと同じであることを確認します。
- 20 サーバー / Web サーバーをクリックし、右のウィンドウで、Web サーバー名の横にあるチェックボックスを選択します。「開始」をクリックします。

**注意：**上記手順オブジェクトウィンドウ実行する前に、リモートマシン (IBM HTTP サーバーマシン) で IBM HTTP 管理サーバーが稼働していることを確認してください。
- 21 Web ブラウザーを開いて、Web サーバーコンピューターの管理コンソール (`http://[web server name]:80/adminui`) にアクセスし、プラグインが正常に生成され、伝播されたかどうかを確認します。次の応答が表示される場合は、手順 21 ~ 24 の説明に従ってプラグインを生成および伝播する必要があります。

```
/[application name] not defined
```

**注意：**プラグインが自動的に生成および伝播されるのは、システムで自動同期が事前に有効になっている場合のみです (デフォルトでは無効)。
- 22 WebSphere Administrative Console にログインし、ナビゲーションツリーで **Servers / Web servers** をクリックし、右側のウィンドウで、HTTP サーバー名の横にある「Select」チェックボックスを選択します。
- 23 「Generate Plug-in」をクリックします。Plugin-cfg.xml ファイルが正常に生成されたことを確認するメッセージが表示されます。
- 24 「Propagate Plug-in」をクリックします。Plugin-cfg.xml ファイルが正常に伝播されたことを確認するメッセージが表示されます。
- 25 Web サーバーを再起動します。

## 第7章：付録 - デプロイヤーのヒープサイズの拡大 (WebSphere 用)

タイムアウトエラーを回避するために、ejbdeploy.bat/sh スクリプトでヒープサイズを拡大する必要があります。

### AIX, Linux, Solaris

- 1 [appserver root]/deploytool/itp/ ディレクトリに移動して、ejbdeploy.sh を編集用に開きます。
- 2 (Solaris のみ) SunOS セクションで、EJBDEPLOY\_JVM\_OPTIONS 属性を探して -XX:PermSize オプションの値を 256m に変更し、-Xverify オプションの値が none であることを確認します。
- 3 \$JAVA\_CMD\ セクションのヒープサイズを以下の値に変更します。  
`-Xms256m Xmx4096m`
- 4 ファイルを保存して閉じます。  
ウィンドウ

### ウィンドウ

- 1 [appserver root]\deploytool\itp に移動して、ejbdeploy.bat ファイルをテキストエディターで開きます。
- 2 %JAVA\_HOME% で始まる行を探して、引数 -Xmx を探します。
- 3 この引数を -Xmx4096M に変更します。
- 4 ファイルを保存して閉じます。

### MaxPermSize の拡大 (Solaris 上の WebSphere)

- 1 WebSphere Administrative Console にログインします。
- 2 WebSphere Administrative Console のナビゲーションツリーで、次のいずれかを実行します。
  - **Servers / Server Types / WebSphere Application servers** をクリックし、右側のウィンドウでサーバー名をクリックします。
- 3 「Server Infrastructure」で、**Java and forms workflow / Process Definition** をクリックします。
- 4 「Additional Properties」で、「**Java Virtual Machine**」をクリックします。
- 5 「**Generic JVM Arguments**」に、MaxPermSize パラメーターとして次のように入力します。  
`-XX:MaxPermSize=1024m`
- 6 「OK」または「Apply」をクリックします。
- 7 「Messages」ボックスで「**Save directly to master configuration**」をクリックし、アプリケーションサーバーを再起動します。