



# **JBoss Enterprise SOA Platform 5**

## **JBoss SOA BPEL ガイド**

JBoss 開発者およびビジネスルール記述者向け  
エディション 5.2.0



# JBoss Enterprise SOA Platform 5 JBoss SOA BPEL ガイド

---

JBoss 開発者およびビジネスルール記述者向け  
エディション 5.2.0

JBoss Riftsaw Development Team コミュニティの協力

edited by the Red Hat Documentation Group

## 法律上の通知

Copyright © 2011 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

本書は、JBoss Enterprise SOA Platform にて BPEL Console をお使いになる開発者向けに作成されています。

## 目次

<b>第1章 サンプル</b> .....	<b>4</b>
1.1. 概要	4
1.2. QUICKSTART のサンプル - HELLO WORLD	4
1.3. ESB サンプル - HELLO WORLD の ESB 呼び出し	4
<b>第2章 管理</b> .....	<b>6</b>
2.1. 概要	6
2.2. BPEL CONSOLE	6
2.2.1. 概要	6
2.2.2. ログイン	6
2.2.3. デプロイしたプロセス定義	6
2.2.4. プロセスインスタンス	6
2.2.5. プロセスバージョン	6
2.2.6. 実行履歴	7
2.2.6.1. ロギング設定オプション	7
2.2.7. インスタンスデータと実行パス	8
2.2.8. プロセス定義のリタイアおよび再有効化	8
2.3. BPEL プロパティ	9
2.4. 実行イベントの有効化	9
2.5. プロセスの安全なアンデプロイ手順	10
<b>第3章 BPEL プロセスのデプロイ</b> .....	<b>11</b>
3.1. 概要	11
3.2. JBOSS SOA PLATFORM SERVER への直接デプロイメント	11
3.3. JBDS ベースのデプロイメント	12
3.4. エンドポイント設定プロパティの変更	13
<b>第4章 WEB サービスの設定</b> .....	<b>15</b>
4.1. 概要	15
4.2. JAX-WS ハンドラーの設定	15
4.3. APACHE CXF 設定	16
4.3.1. サーバーのエンドポイントの設定	16
4.3.2. クライアントのエンドポイントの設定	17
4.4. デプロイされたサービス WSDL	19
<b>第5章 UDDI 統合</b> .....	<b>20</b>
5.1. 概要	20
5.2. UDDI 設定プロパティ	20
5.3. その他の UDDI V3 レジストリ	22
5.4. UDDI レジストリエンティティと UDDI SEED データ	22
<b>第6章 JBOSS ENTERPRISE SOA PLATFORM 統合</b> .....	<b>24</b>
6.1. 概要	24
6.2. BPELINVOKE ESB アクションの利用	24
6.2.1. フォルト処理	26
6.2.2. SAML サポート	27
<b>第7章 クラスターリングサポート</b> .....	<b>28</b>
7.1. インストール	28
7.2. 設定	28
7.3. BPEL プロセスサービスの呼び出し	29
<b>付録A GNU GENERAL PUBLIC LICENSE 2.0</b> .....	<b>30</b>

付録B 改訂履歴 ..... 37



# 第1章 サンプル

## 1.1. 概要

本章は、サンプルのスタートガイドとなっており、個別のサンプルに関する詳細情報は、サンプルフォルダーを参照してください。

## 1.2. QUICKSTART のサンプル - HELLO WORLD

「Hello world」のサンプルが最もシンプルです。これは、**SOA\_ROOT/samples/quickstarts/hello\_world** ディレクトリに置かれています。

まず、このサンプルを実行中の JBoss サーバーにデプロイします。方法としては、**ant deploy** のコマンドを使いデプロイします。

**ant** スクリプトから正常に終了したとのメッセージが出されると、次のステップに進み、先ほどデプロイした BPEL プロセスを呼び出します。

提供された **ant** スクリプト **ant sendhello** を実行すると、BPEL プロセスを呼び出すことができます。

サンプルメッセージは、**messages** サブディレクトリにあります。

メッセージが BPEL プロセスに渡されると、レスポンスが返ってきます。

サンプルを変更したり、新規バージョンをデプロイしたりする場合は、**build.xml** の 'version' プロパティを変更するか、**ant -Dversion=2 deploy** のようにコマンド実行時にバージョンを指定してください。

アンデプロイするには、**ant -Dversion=2 undeploy** のコマンドを実行します。



### 注記

過去にデプロイした BPEL デプロイメントを変更するには、以前のバージョンをアンデプロイするか、バージョン番号を1つあげる必要があります。すでにデプロイしたユニットと同じ名前でも BPEL デプロイメントユニットをデプロイしようとすると、サーバーが例外を発行します。



### 警告

現在、バージョンは整数値で指定する必要があります。数値でない **major.minor.incremental** (maven スタイル) のような形式のバージョンであれば、サーバーをデプロイすると例外が発生します。

## 1.3. ESB サンプル - HELLO WORLD の ESB 呼び出し

エンジンと ESB が同じ JVM に置かれており、呼び出される BPEL プロセスが ESB と共存する BPEL エンジンインスタンスにデプロイされる限り、ESB サービスがエンジン内で実行中の BPEL プロセスを直



接呼び出すことができるのですが、この **quickstart** のサンプルは、その仕組みについて説明していません。

サンプルは、**SOA\_ROOT/samples/esb/bpel\_helloworld** ディレクトリに置かれています。また、前章で説明されている **quickstart (SOA\_ROOT/samples/quickstarts/hello\_world)** がまだデプロイされている点に注意してください。このサンプルがデプロイされていない場合は、前章の説明にしたがい、このサンプルをデプロイしてください。

次に、**"hello world"** BPEL プロセスを呼び出す ESB サービスをデプロイします。**SOA\_ROOT/samples/esb/bpel\_helloworld** ディレクトリで、**ant deploy** のコマンドを実行することで、デプロイ可能です。

ESB サービスが正常にデプロイされると、**ant runtest** のコマンドを実行します。

このコマンドは、ESB 経由で **"Hello World via ESB to BPEL"** という文字を ESB サービスに送り BPEL プロセスが呼び出されます。すると、この文字に **"World"** が追記され、クライアントアプリケーションが受け取るまで **echo** を返し続けます。

## 第2章 管理

### 2.1. 概要

BPEL Console の管理機能については、本章を参照してください。

### 2.2. BPEL CONSOLE

#### 2.2.1. 概要

BPEL Console は、以下を確認することのできるツールです。

- BPEL エンジンにデプロイしたプロセス定義
- BPEL エンジンで実行する プロセスインスタンス
- プロセス実行履歴

#### 2.2.2. ログイン

BPEL Console にアクセスするには、Web ブラウザーを起動して <http://localhost:8080/bpel-console> に移動します。そうすると、ログイン画面が表示されます。

デフォルトユーザーは **admin**、パスワードも **admin** となっています。それぞれ、**soa-user.properties** と **soa-roles.properties** ファイルに定義されています。

ログインポリシーは **login-config.xml** ファイルに定義されています。**deploy/bpel-console/gwt-console-server.war/WEB-INF/jboss-web.xml** ファイルの **<security-domain/>** 要素を使い変更可能です。

#### 2.2.3. デプロイしたプロセス定義

正常にログインすると、**Manage Instances** をクリックして現在デプロイされている BPEL プロセスを表示します。各プロセスのバージョン情報も表示されます。

#### 2.2.4. プロセスインスタンス

プロセス定義を1つ選択して開きます。一番下のパネルで、その定義で有効なプロセスインスタンス一覧が表示されます。

#### 2.2.5. プロセスバージョン

1回につき、プロセスのバージョン1つのみが有効です。プロセス定義を開くと、アクティブなバージョンが自動的に選択されます。

古いバージョンを管理する必要がある場合もあります (実行中のインスタンスを終了させるなど)。このような場合は、**More - Change Version** をクリックして、希望のバージョンを選択します。

**Quickstart\_bpel\_simple\_invoke.jar** などのように、プロセスアーカイブにバージョンがない場合、バージョンゼロとして処理されます (**Quickstart\_bpel\_simple\_invoke-1.jar**, が次にデプロイされるバージョンです)。

## 2.2.6. 実行履歴

プロセスに関連付けられた **BPAF (Business Process Analytics Format)** 履歴データを確認したい場合、**Execution History** でチャートを生成します。一定のレビュー期間を指定して問題のあるインスタンスや終了したインスタンスを選択しチャートに含めることができます。

以下のショートカットキーを用いてチャート内を移動します。

表2.1 Aggregator プロパティ

キーボードまたはマウスのコマンド	結果
上向き矢印	拡大
下向き矢印	縮小
左向き矢印	左側半分
右向き矢印	右側半分
Page-Up	左ページへ
Page-Down	右ページへ
TAB	次のアイテムへ
Shift-TAB	以前のアイテムへ
HOME	最大に拡大
ENTER	最大限に縮小してフォーカス
マウスでドラッグ	チャートのスクロール
Shift を押しながらマウスでドラッグ	選択したものをドラッグ／拡大
Mouse Wheel Up/Z	拡大
Mouse Wheel Down/X	縮小
Backspace/Back ボタン	戻る
右クリック	コンテキストメニュー
左クリック	フォーカス設定
ダブルクリック	最大限に拡大

### 2.2.6.1. ロギング設定オプション

BPEL Console のロギング機能を有効にするには、**deploy.xml** 設定ファイルと、**bpel.properties**

ファイルの2つの箇所では有効化する必要があります。

ロギングしたい特定のプロセスに **process-events** オプションを付け、**org.jboss.soa.bpel.console.bpaf.BPAFLogAdapter** が有効になっていることを確認します。

方法を示した **deploy.xml** サンプルファイルを以下に示しています。

### 例2.1 サンプル例

```
<deploy xmlns="http://www.apache.org/ode/schemas/dd/2007/03"
  xmlns:bp1="http://www.jboss.org/bpel/examples"
  xmlns:intf="http://www.jboss.org/bpel/examples/wsdl">

  <process name="bp1:HelloGoodbye">
    <active>true</active>
    <process-events generate="all"/>
    <provide partnerLink="helloGoodbyePartnerLink">
      <service name="intf:HelloGoodbyeService" port="HelloGoodbyePort"/>
    </provide>
  </process>
</deploy>
```

## 2.2.7. インスタンスデータと実行パス

プロセスインスタンスを選択すると、**Execution Details** パネルにプロセスインスタンスの詳細が表示され、**Instance Data** ボタンが有効になります。このボタンをクリックして、プロセスの詳細を参照します。または、**Execution Path** ボタンをクリックして、プロセスの *instance execution graph* を表示することも可能です。

## 2.2.8. プロセス定義のリタイアおよび再有効化

BPEL プロセス定義の最初のバージョンをデプロイすると、自動的にアクティブな定義になります。この定義が後に変更され、再度デプロイされると、そのバージョンは「リタイア」し、新しいバージョンが自動的にアクティブになります。

アクティブなバージョンと、リタイアしたバージョンの違いは、リタイアしたバージョンでは新しいプロセスインスタンスを作成できなくなっていることです。しかし、リタイアしたプロセスバージョンに関連付けられたアクティブなプロセスインスタンスがある場合、このプロセスインスタンスは継続して実行されます。

プロセス定義のステータスを変更したい場合があります。変更するには、**Runtime** タブ (左側のパネル) をクリックして、**Deployments** オプションを選択します。各プロセス定義のバージョン情報と現在のステータス (**active** (アクティブ) または **retired** (リタイア)) を表示することができます。

**retired** から **active** へ特定のバージョンを変更するには、リタイアしたバージョンを選択して **Activate** ボタン (画面の右下) を押します。

アクティブなプロセス定義をリタイアさせるには、リタイアしたいバージョンを選び、**Retire** ボタンを押します。



## 注記

プロセスをアンデプロイする場合、プロセスの以前のバージョンがなければエンドポイントのみが無効になります。

## 2.3. BPEL プロパティ

JBoss Enterprise SOA Platform に対して BPEL Console を設定するには、`/server/<profile>/deploy/riftsaw.sar/bpel.properties` ファイルを編集する必要があります。

この図は、設定可能な BPEL Console 固有のプロパティを示しています。

表2.2 BPEL 固有のプロパティ

UDDI プロパティ	
<code>bpel.jaxws.client.initializer.impl</code>	このプロパティは、使用する JAXWS スタックに基づき、インストール時に自動的に設定されます。この値は変更しないでください。
<code>bpel.ws.stableInterface</code> (デフォルト <code>false</code> )	このプロパティは、BPEL プロセスに関連付けられた Web サービスインターフェースが BPEL プロセスの新バージョンのデプロイ時に更新されるかを決定します。この項目を <code>false</code> に設定する利点は、WSDL への変更が BPEL プロセスにあわせてアクティブになる点です。ただし、Web サービスが一時的に利用できなくなり、使用頻度の非常に高いサービスがリクエストを拒否する可能性があるため、インターフェース間の移行時に問題がでてきます。この値を <code>true</code> に設定すると、Web サービスは BPEL プロセスの更新時も利用できますが、WSDL への変更は利用できません。
<code>bpel.event.listeners</code> ( <code>org.jboss.soa.bpel.console.bpaf.BPAFLogAdapter</code> )	コンソールで利用する BPAF ロギングを有効にしてプロセス実行履歴を表示します。



## 注記

このファイルには、**Apache ODE** 固有のプロパティが多数含まれています。このプロパティの詳細について確認されたい場合は、**ODE** ドキュメンテーションを参照してください(このファイルのプロパティ名にプレフィックス `bpel` がついている可能性がありますのでご注意ください。**ODE** では、プレフィックスは `ode` となっています)。

## 2.4. 実行イベントの有効化

**Apache ODE** を有効にすると、実行中のプロセスに対してイベントセットを生成します。以下の手順に従ってください。

1. `<process-events generate="all"/>` を `deploy.xml` に追加します。
2. `bpel.event.listeners=org.apache.ode.bpel.common.evt.DebugBpelEventListener` と `debugeventlistener.dumpToStdOut=on` を `bpel.properties` ファイルに追加します。

`DebugBpelEventListener` を登録することで、`BpelEvent` 情報に関するログの **DEBUG** レベルを  
使えるように有効化します。`debugeventlistener.dumpToStdOut=on` を設定することで、  
情報を **Console** にも出力できるようになります。



### 注記

カスタムリスナーを追加する方法については、`bpel_event_listener quick-start`  
を参照してください。

## 2.5. プロセスの安全なアンデプロイ手順

アクティブなプロセス定義をアンデプロイすると、そのプロセス定義に所属するプロセスインスタンス  
がすべて終了されます。終了したくない場合は、デプロイ前にプロセス定義をリタイアしてください。

手順を繰り返す場合も、プロセスを安全にアンデプロイする方法に従うようにしてください。

1. プロセスにアクティブなプロセスインスタンスがあるかどうか確認します。
2. ある場合は、**BPEL Console** を使い、まずそのプロセスインスタンスをリタイアします。
3. アクティブなプロセスインスタンスがなくなってから、プロセスをアンデプロイしてください。

## 第3章 BPEL プロセスのデプロイ

### 3.1. 概要

プロセスを BPEL Console にデプロイする方法については、本章を参照してください。

### 3.2. JBOSS SOA PLATFORM SERVER への直接デプロイメント

JBoss Developer Studio で BPEL プロセスを直接テストすることができます。Ant スクリプトは必要ありません。プロセスを右クリックして、**WSDL** から **Web Services**、**Test with Web Service Explorer** を選択します。

一方、サンプルのクイックスタートは直接デプロイメントについて例示しています。このメソッドに使用される ant スクリプトの種類に関する例を以下に示します。

#### 例3.1 サンプル例

```
<!-- Import the base Ant build script... -->
<property file="../../install/deployment.properties" />

<property name="version" value="1" />

<property name="server.dir"
value="${org.jboss.as_home}/server/${org.jboss.as_config}"/>
<property name="conf.dir" value="${server.dir}/conf"/>
<property name="deploy.dir" value="${server.dir}/deploy"/>
<property name="server.lib.dir" value="${server.dir}/lib"/>

<property name="sample.jar.name" value="${ant.project.name}-
${version}.jar" />

<target name="deploy">
  <echo>Deploy ${ant.project.name}</echo>
  <jar basedir="bpe1" destfile="${deploy.dir}/${sample.jar.name}" />
</target>

<target name="undeploy">
  <echo>Undeploy ${ant.project.name}</echo>
  <delete file="${deploy.dir}/${sample.jar.name}" />
</target>
```

以下に主なポイントを挙げています。

- BPEL プロセスがデプロイされる JBoss SOA Platform Server の場所を提供する必要があります (この例では、ディストリビューションで設定 (installation ディレクトリ) されている **deployment.properties** ファイルを参照することで JBoss SOA Platform Server の場所を提供できます)。
- バージョンアップロード (同じ BPEL プロセスの複数バージョンを同時にデプロイする方法) をとっている場合、BPEL プロセスと関連のアーチファクトを含む **JAR** アーカイブファイル名にバージョン番号のサフィックスが付きます。別のバージョンの BPEL プロセスをデプロイするたびに、このサフィックスの番号を手動で1つ増やす必要があります。



### 警告

現在、バージョンは整数で示す必要があります。**major.minor.incremental (Maven スタイル)** などのように、数値以外の値の場合は例外がトリガーされます。

- 次は、BPEL プロセスアーカイブを作成する際に利用するデプロイメントの対象を定義する必要があります。今回は、**bpel** サブディレクトリの内容を使い、**JBoss Enterprise SOA Platform** の **deploy** ディレクトリに保存します。
- 最後に、アンデプロイの対象を定義する必要があります。これは、**JBoss Enterprise SOA Platform** の **deploy** ディレクトリから BPEL プロセスアーカイブを削除するために利用するだけです。

## 3.3. JBDS ベースのデプロイメント

この説明に従い、**JBoss Enterprise SOA Platform** で実行中の JBDS BPEL プロジェクトを BPEL Console にデプロイします。

1. まず、JBDS BPEL プロジェクトを作成またはインポートします。この例では、**\$/samples/quickstart/hello\_world** ディレクトリから既存プロジェクトをインポートしています。

JBFS の左側のナビゲーターパネルにある **Import** メニューを選択します。次に **General->Existing Projects into Workspace** エントリを選び、**Next** ボタンをクリックしてください。

2. **Browse** ボタンをクリックして、**hello\_world quick start** ディレクトリへ移動します。ディレクトリに移動してから、**Finish** ボタンを押します。

プロジェクトがインポートされると、BPEL プロセスや WSDL の説明など、コンテンツを確認することができます。

3. **JBoss SOA Platform** のサーバー設定を作成します。**Server** タブが JBDS ウィンドウの下側に表示されるはずですが、表示されていない場合、**Window->Show Views->Servers** メニューへ移動し、手動でビューを開いてください。
4. **Servers** ビューで右クリックして、**New->Server** メニューを選択します。
5. **JBoss SOA Platform** の適切なバージョンを選択して、**Finish** を押します。
6. 新しいサーバーを起動します。**Servers** タブにあるサーバーを右クリックして、**Start** メニューをクリックすると起動できます (**Console** タブにサーバーからの出力が表示されます)。
7. サーバーが開始されると、サーバーエントリを右クリックし、**Add and Remove** メニューを選択します。
8. **Quickstart\_bpel\_hello\_world** プロジェクトを選択して、**Add** ボタンを押してから **Finish** ボタンを押します。このプロジェクトがサーバーにデプロイされます。
9. プロジェクトがデプロイされると、**Servers** タブのサーバーの下に表示されます。

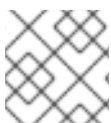


10. デプロイされた BPEL プロセスを **quickstart** のサンプルにある **Ant** スクリプトを使ってテストするには、プロジェクトの **root** フォルダの **build.xml** ファイルを右クリックして、**Run As->Ant Build...** メニューを選択します ("..." のついたメニューアイテムを選択することが重要です。ダイアログボックスが表示され、使いたい **Ant** を選択できるようになるからです)。
11. **deploy** ターゲットの選択を解除して、**sendhello** ターゲットを選択し、**Run** ボタンをクリックします。'hello' というテストメッセージがサーバーに送信され、レスポンスが **Console** タブに表示されます。
12. BPEL プロセスを更新したい場合は、図の中の **assignHelloMesg** ノードを選択し、**Properties** ビューを選びます。ビューの左側に縦向きのタブが表示されています。**Details** タブを選んでから、一覧内の **Expression to Variable** を選択して、**UPDATED** をテキストに追加するために **concat** 機能の 2 番目のパラメーターを更新します。
13. 更新を保存すると、**Server** ビューへ移動して、**Quickstart\_bpel\_hello\_world** プロジェクトに関連付けられたメニューから **Full Publish** オプションを選択します。このプロジェクトがサーバーに再デプロイされます。
14. 最後のステップでは、新たなリクエストを送り、このレスポンスを表示するため、**build.xml** ファイル内の **sendhello** ターゲットを再度実行します。このレスポンスは、BPEL プロセスで変更を加えたとおりに変更されているはずです。

**Server** ビューに移動し、デプロイしたプロジェクトノードを展開すると、デプロイした両バージョンが表示されます (旧バージョンを利用している BPEL プロセスインスタンスがまだ残っている可能性もあるため、旧バージョンは維持されます)。子ノードを右クリックすると、各バージョンを別でアンデプロイできることがわかります。しかし、バージョンを手動でアンデプロイすると、そのバージョンで有効な残りのプロセスインスタンスが終了されます。

**Server** ビューにある、プロジェクトに関連付けられたメニューを使い、**(Add and Remove メニューで)** プロジェクトをアンデプロイし、最終的にサーバーに関連付けられたメニューを使いサーバーを停止します。

### 3.4. エンドポイント設定プロパティの変更



#### 注記

BPEL Console は現在 **mex.timeout** のみをサポートしています。

本章の残りを参照して、これらの **.endpoint** ファイルのデプロイ方法を学習します。



#### 注記

BPEL Console は現在グローバル設定機能に対応していないため、デプロイした BPEL バンドルからしか設定データを取得できない点にご留意ください (具体的には、BPEL デプロイメントユニット内の **root** に、BPEL の配備記述子とあわせて表示されています)。

**/samples/quickstart/hello\_world/bpelContent** ディレクトリに **test.endpoint** と呼ばれるファイルを置いて練習します。以下のコンテンツが含まれているようにしてください。

#### 例3.2 サンプル例

```
# 3 minutes
mex.timeout=180000
```

デプロイされると、**helloworld** のサンプルでは、レスポンスの最大待機時間は **3 分間** となっています。テストするには、**hello\_world.bpel** ファイルを編集して、レスポンスの前に **wait** アクティビティを追加します。

### 例3.3 サンプル例

```
<wait>
  <for>'PT150S'</for>
</wait>
```

ここでは、応答前に **2 分 30 秒** 待機します。デフォルトのタイムアウトより **30 秒** 長くなっていますが、**test.endpoint** ファイルで指定した新しいタイムアウト時間内となっています。タイムアウトを強制的に起こさせたい場合は、待機時間を **3 分 30 秒** に増やし、**ant sendhello** コマンドを使いテストメッセージを再度サブミットしてください。

## 第4章 WEB サービスの設定

### 4.1. 概要

Web サービススタックの設定に利用可能な仕組みについては、本章を参照してください。このスタックを使い、Web サービスを BPEL プロセスに渡します。本章を学習すると、BPEL プロセスから外部の Web サービスを呼び出す方法も学習することができます。

### 4.2. JAX-WS ハンドラーの設定

JAX-WS は標準 Java API で、Web サービスサポートを提供します。Web サービスが JAX-WS ハンドラーの仕組みを用いて、メッセージ (問題) の送受信時にユーザーが指定するクラスを呼び出します。そのため、このハンドラーはメッセージのパイプラインにインストールされ、必要に応じてこれを使いメッセージヘッダーやボディを操作します。

通常、プログラムで Web サービスを表す Java インターフェースにハンドラーをインストールするか、**HandlerChain** アノテーションでインストールします。ただし、BPEL にデプロイされるプロセスの場合、JAX-WS サービスはデプロイ時に動的に作成されます。

そのため、JAX-WS ハンドラーチェーンの設定と、BPEL プロセス対応のためにデプロイされるよう動的に作成された Web サービスを関連付けるには、BPEL プロセスの定義と配備記述子と一緒に **jws\_handler.xml** と呼ばれるファイルを置く必要があります。

以下に **jws\_handler.xml** ファイルの例を挙げています。この例は、**service\_handler quick-start** で使用しています。

```
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>JAXWSHandler</handler-name>
      <handler-class>org.jboss.soa.bpel.examples.jaxws.JAXWSHandler</handler-
class>
      <init-param>
        <param-name>TestParam</param-name>
        <param-value>TestValue</param-value>
      </init-param>
    </handler>
  </handler-chain>
</handler-chains>
```

このファイルは、標準の JAX-WS ハンドラーチェーン設定形式です。名前とクラスを定義するハンドラーをそれぞれ用いることで、ハンドラー要素は1つでも、複数でも指定できます。任意でハンドラー設定は、ハンドラー実装の **init** メソッドに渡す初期化パラメーターを提供します。



#### 注記

この仕組みは、JAX-WS ハンドラーをプロバイダー Web サービスにインストールします。現在、BPEL プロセスから外部の Web サービスを呼び出す際に利用するクライアントのエンドポイントには、JAX-WS ハンドラーを設定できません。

この仕組みのサンプルについては、**service\_handler quick-start** を参照してください。

### 4.3. APACHE CXF 設定

BPEL は JAX-WS API を使用して JBossWS と統合します。これにより、BPEL が JBossWS ネーティブと Apache CXF Web サービススタックの両方をサポートできるようになります。本章は、BPEL プロセスに追加の設定情報を含める方法、特に Apache CXF Web サービススタック関連について説明します。

#### 4.3.1. サーバーのエンドポイントの設定

サーバーが利用できるように CXF 設定を作成するには、BPEL デプロイメントの root ディレクトリに **jbossws-cxf.xml** と呼ばれるファイル (配備記述子と合わせて) を設置してください。

JAX-WS アノテーションベースの Web サービスをデプロイする場合は、**jbossws-cxf** で使用するファイルと同じです。このファイルの内容の例を以下に示しています。

```
<beans
  xmlns='http://www.springframework.org/schema/beans'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:beans='http://www.springframework.org/schema/beans'
  xmlns:jaxws='http://cxf.apache.org/jaxws'
  xsi:schemaLocation='http://cxf.apache.org/core
    http://cxf.apache.org/schemas/core.xsd
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
    http://cxf.apache.org/jaxws
    http://cxf.apache.org/schemas/jaxws.xsd'>

  <bean id="UsernameTokenSign_Request"
    class="org.apache.cxf.ws.security.wss4j.WSS4JInInterceptor">
    <constructor-arg>
      <map>
        <entry key="action" value="UsernameToken Timestamp Signature"/>
        <entry key="passwordType" value="PasswordDigest"/>
        <entry key="user" value="serverx509v1"/>
        <entry key="passwordCallbackClass"

value="org.jboss.test.ws.jaxws.samples.wsse.ServerUsernamePasswordCallback
"/>
        <entry key="signaturePropFile"
value="etc/Server_SignVerf.properties"/>
        <entry key="signatureKeyIdentifier" value="DirectReference"/>
      </map>
    </constructor-arg>
  </bean>

  <bean id="UsernameTokenSign_Response"
    class="org.apache.cxf.ws.security.wss4j.WSS4JOutInterceptor">
    <constructor-arg>
      <map>
        <entry key="action" value="UsernameToken Timestamp Signature"/>
        <entry key="passwordType" value="PasswordText"/>
        <entry key="user" value="serverx509v1"/>
        <entry key="passwordCallbackClass"

value="org.jboss.test.ws.jaxws.samples.wsse.ServerUsernamePasswordCallback
"/>
        <entry key="signaturePropFile"
```

```

value="etc/Server_Decrypt.properties"/>
    <entry key="signatureKeyIdentifier" value="DirectReference"/>
    <entry key="signatureParts"
        value="{Element}{http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd}Timestamp;{Element}
{http://schemas.xmlsoap.org/soap/envelope/}Body"/>
    </map>
</constructor-arg>
</bean>

<jaxws:endpoint
    id='SecureHelloWorldWS'

address='http://@jboss.bind.address@:8080/Quickstart_bpel_secure_serviceWS'
,
    implementor='@provider@'>
<jaxws:inInterceptors>
    <ref bean="UsernameTokenSign_Request"/>
    <bean class="org.apache.cxf.binding.soap.saaj.SAAJInInterceptor"/>
</jaxws:inInterceptors>
<jaxws:outInterceptors>
    <ref bean="UsernameTokenSign_Response"/>
    <bean
class="org.apache.cxf.binding.soap.saaj.SAAJOutInterceptor"/>
</jaxws:outInterceptors>
</jaxws:endpoint>

</beans>

```

このサンプルコードは、Web サービスがユーザー名 **token** とデジタル署名認証を合わせて利用できるように設定しています。



### 注記

**jaxws:endpoint** 要素には、**implementor** と呼ばれる属性があります。この属性は、JAX-WS サービスを実装する Java クラスを定義します。BPEL Console は、このクラスを動的に自動作成します。そのため、この属性の値は **@provider@** にしなければ機能しません。

### 4.3.2. クライアントのエンドポイントの設定

BPEL を呼び出す Web サービスを表すクライアントエンドポイントは、ポート別にそれぞれ違うファイルで設定されており、各種設定が分散しています。

これらのファイルは、**jbossws-cxf-{portname\_local\_part}.xml** のような名前のルールに従います。**portname\_local\_part** の部分は呼び出される Web サービスのポート名のローカル部を指します。WSDL のサンプルを以下に示します。

```

<definitions name='SecureHelloWorldWSService'
    targetNamespace='http://secure_invoke/helloworld' .... >
    <portType name='SecureHelloWorld'>
        ...
    </portType>

```

```

<service name='SecureHelloWorldWSService'>
  <port name='SecureHelloWorldPort' ... >
    ...
  </port>
</service>
</definitions>

```

この場合、CXF 設定ファイルは **jbossws-cxf-SecureHelloWorldPort.xml** となります。

このファイルに含まれる設定情報は、CXF バス用です。以下に例を示します。

```

<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cxf="http://cxf.apache.org/core"
  xmlns:wsa="http://cxf.apache.org/ws/addressing"
  xmlns:http="http://cxf.apache.org/transports/http/configuration"
  xmlns:wsm-policy="http://schemas.xmlsoap.org/ws/2005/02/rm/policy"
  xmlns:wsm-mgr="http://cxf.apache.org/ws/rm/manager"
  xmlns:beans='http://www.springframework.org/schema/beans'
  xmlns:jaxws='http://cxf.apache.org/jaxws'
  xmlns:ns1='http://secure_invoke/helloworld'
  xsi:schemaLocation="
    http://cxf.apache.org/core http://cxf.apache.org/schemas/core.xsd
    http://cxf.apache.org/transports/http/configuration
    http://cxf.apache.org/schemas/configuration/http-conf.xsd
    http://schemas.xmlsoap.org/ws/2005/02/rm/policy
    http://schemas.xmlsoap.org/ws/2005/02/rm/wsm-policy.xsd
    http://cxf.apache.org/ws/rm/manager
    http://cxf.apache.org/schemas/configuration/wsm-manager.xsd
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://cxf.apache.org/jaxws http://cxf.apache.org/schemas/jaxws.xsd">

  <bean id="UsernameTokenSign_Request"
    class="org.apache.cxf.ws.security.wss4j.WSS4JOutInterceptor" >
    <constructor-arg>
      <map>
        <entry key="action" value="UsernameToken Timestamp Signature"/>
        <entry key="passwordType" value="PasswordDigest"/>
        <entry key="user" value="clientx509v1"/>
        <entry key="passwordCallbackClass"
value="org.jboss.test.ws.jaxws.samples.wsse.ClientUsernamePasswordCallback
"/>
        <entry key="signaturePropFile"
value="etc/Client_Sign.properties"/>
        <entry key="signatureKeyIdentifier" value="DirectReference"/>
        <entry key="signatureParts"
value="{Element}{http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd}Timestamp;{Element}
{http://schemas.xmlsoap.org/soap/envelope/}Body"/>
        </map>
      </constructor-arg>
    </bean>

  <bean id="UsernameTokenSign_Response"

```

```

        class="org.apache.cxf.ws.security.wss4j.WSS4JInInterceptor" >
<constructor-arg>
    <map>
        <entry key="action" value="UsernameToken Timestamp Signature"/>
        <entry key="passwordType" value="PasswordText"/>
        <entry key="user" value="serverx509v1"/>
        <entry key="passwordCallbackClass"
value="org.jboss.test.ws.jaxws.samples.wsse.ClientUsernamePasswordCallback
"/>
        <entry key="signaturePropFile"
value="etc/Client_Encrypt.properties"/>
        <entry key="signatureKeyIdentifier" value="DirectReference"/>
    </map>
</constructor-arg>
</bean>

<cxf:bus>
    <cxf:outInterceptors>
        <ref bean="UsernameTokenSign_Request"/>
        <bean
class="org.apache.cxf.binding.soap.saaj.SAAJOutInterceptor"/>
    </cxf:outInterceptors>
    <cxf:inInterceptors>
        <ref bean="UsernameTokenSign_Response"/>
        <bean class="org.apache.cxf.binding.soap.saaj.SAAJInInterceptor"/>
    </cxf:inInterceptors>
</cxf:bus>

</beans>

```

これらの設定は、Web サービスクライアントが **token** ユーザー名とデジタル署名を合わせて利用できるよう設定します。

## 4.4. デプロイされたサービス WSDL

BPEL プロセスをデプロイすると、Web サービスは自動的に作成されます。この Web サービスはサービスのエンドポイントを表し、デプロイ中のプロセスの一部として含まれる WSDL の記述をベースとしています。

**wsdl** サフィックスを使用して、サービスのエンドポイント URL 経由で WSDL を表示できます。デフォルトでは、WSDL が一旦表示されると、**<soap:address>** はサーバーのバインドアドレスと関連付けられます (このアドレス自体は、**\$(jboss.bind.address)** プロパティで定義されます)。

# 第5章 UDDI 統合

## 5.1. 概要



**重要**

BPEL Console が **JBoss Enterprise SOA Platform** にデプロイされると、jUDDI はインストールされないため、UDDI 統合がオフになります (`bpel.uddi.registration=false`)。

BPEL は *UDDI* レジストリが統合されているため、サービスはデプロイされると自動的に登録されます。

この登録プロセスは、jUDDI-3 クライアントライブラリを使います。サービスがデプロイされると、プロセスとプロセスの *BindingTemplate* が登録され、*partnerLinkChannel* が *partnerLink* ごとに作成されます。同時に WSDL エンドポイントが UDDI から取得されます (この URL を使い WSDL をサービスから直接取得します)。

アンデプロイすると、*BindingTemplate* が **UDDI Registry** から削除されます。

## 5.2. UDDI 設定プロパティ

BPEL Console は、**JBoss Enterprise SOA Platform** の jUDDI クライアントライブラリを使います。クライアント設定ファイルは `deploy/jbossesb.sar/esb.juddi.client.xml` です。



**注記**

*ClerkManager* と *Clerk* の名前が `bpel.properties` ファイルで指定されます。

表5.1 `bpel.properties` ファイルの UDDI 関連のプロパティ

属性	タイプ (デフォルト)	説明
<code>bpel.uddi.registration</code>	boolean (true)	False に設定されると、UDDI 統合はオフになります。jbossesb-registry.sar が jUDDI v3 レジストリを含むと検出した場合のみ、RiftSaw のインストールプロセスでは、この値は true に設定されます。それ以外の場合は、自動的に false に設定されます。



属性	タイプ (デフォルト)	説明
<code>bpel.webservice.secure</code>	boolean (false)	UDDI 登録プロセスは、登録している BPEL サービスに対して、 <b>BindingTemplate</b> に <b>WSDL AccessPoint</b> を登録します。BPEL サービスは、サービス WSDL エンドポイントを WS スタックに公開します (Red Hat は現在 JBossWS と CXF をサポートします)。Web サービススタックがセキュアなプロトコル ( <b>https</b> など) を使うように設定されている場合、この設定を <b>true</b> に設定する必要があります (登録プロセスにのみ、この設定を使います)。
<code>bpel.uddi.client.impl</code>	文字列 ( <code>org.jboss.soa.bpel.uddi.UDDIRegistrationImpl</code> )	これは、 <b>org.jboss.soa.bpel.runtime.engine.ode.UDDIRegistration</b> インターフェースを実装するクラス名です。
<code>bpel.uddi.clerk.config</code>	文字列 (デフォルトでは使用されません)	これは、 <b>bpel.uddi.client.xml</b> 設定ファイルまでのパスを定義します。 <b>jbossesb.sar/esb.uddi.client.xml</b> を使いたい場合は、コメントアウトした状態で残すこともできます。ただし、 <b>bpel.uddi.clerk.manager</b> を定義する必要があります。
<code>bpel.uddi.clerk.manager</code>	文字列 (esb-registry)	これは、 <b>bpel.uddi.clerk.config</b> がコメントアウトのままで残された場合に使用される <b>ClerkManager</b> の名前を定義します。この値は、 <b>esb.juddi.client.xml</b> のマネージャーの名前と一致する必要があります。 <b>bpel.uddi.clerk.config</b> を定義すると、 <b>bpel.uddi.clerk.manager</b> の設定は無視されます。



属性	タイプ (デフォルト)	説明
<code>bpel.uddi.clerk</code>	文字列 (BPEL_clerk)	これは、使用予定の Clerk の名前を定義します。この値は、 <b>esb.juddi.client.xml</b> の Clerk の名前と一致する必要があります (デフォルトでは BPEL_clerk に設定されています)。
<code>bpel.uddi.lookup</code>	boolean (true)	これが <b>true</b> に設定されている場合は、パートナーチャネルの作成プロセスは、UDDI 内を <b>serviceName</b> 別にルックアップして、WSDL エンドポイントをリトリブします。この WSDL エンドポイントを使用して、WSDL を取得します。このプロセスは、BPEL デプロイメントで <b>partnerlink</b> WSDL ファイルを更新する必要なしに、デプロイメント内でエンドポイントを簡単に移動できます。 <b>partnerlink</b> ごとに初期の <b>partnerlink</b> WSDL ファイルをデプロイする必要があります。

### 5.3. その他の UDDI V3 レジストリ

その他の UDDI バージョン 3 に準拠するレジストリを利用することができます。SOAP ベースのコミュニケーションを設定するには、JAXWS-Transport を指定して、ご自身の **bpel.uddi.client.xml** を使用することができます。

### 5.4. UDDI レジストリエンティティと UDDI SEED データ

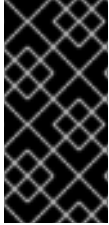
**esb.uddi.client.xml** ファイルでは、登録時に Clerk で使用するプロパティをいくつか定義することができます。しかし、これらの設定をカスタマイズする場合、**jbossesb-registry.sar/juddi\_custom\_install\_data** ファイルの *JBoss ESB Publisher* に指定されている UDDI seed データと一致するようにしてください。

以下にデフォルト設定を挙げます。

表5.2 デフォルト設定

プロパティ	値
<code>keyDomain</code>	<code>esb.jboss.org</code>
<code>businessKey</code>	<code>redhat-jboss</code>

プロパティ	値
serviceDescription	BPEL Service deployed by Riftsaw
bindingDescription	BPEL Endpoint deployed by Riftsaw



### 重要

**jbossesb-registry.sar/esb.uddi.xml** ファイルには、**juddi.seed.always** と呼ばれるプロパティが含まれており、**true** に設定されています。つまり、サーバーが起動すると **root seed** データのロードを試行します。Red Hat は、**seed** データに問題がないと判断した場合は、この値を **false** に変更するよう推奨しています。

## 第6章 JBOSS ENTERPRISE SOA PLATFORM 統合

### 6.1. 概要

BPEL Console と JBoss Enterprise SOA Platform との統合についてさらに理解を深めるには、本章を参照してください。

双方向の疎結合機能は、Web サービスを使うことで利用できます。例えば、ESB アクションは Web サービスを使い BPEL プロセスを呼び出す場合があります。この Web サービスは WSDL インターフェースで表されます。同様に、BPEL プロセスは Web サービス機能ともなるえる ESB 管理サービスを呼び出すことができます。

Web サービスを使わずに BPEL と ESB アクションを統合することもできます。方法については、本章の残りのセクションを参照してください。

### 6.2. BPEL INVOKE ESB アクションの利用

BPELInvoke ESB アクションを `jboss-esb.xml` ファイル内で使用して、BPEL 内で実行中の BPEL プロセスで呼び出しをリクエストします。唯一の制限は、同じ Java 仮想マシン上に BPEL をインストールし、リクエストしたプロセスがローカルな BPEL エンジンにデプロイされている必要があるという点のみです。

以下の例では、`bpel_helloworld quick-start` と合わせて使用するアクションについて説明しています。

#### 例6.1 サンプル例

```
<action name="action2"
class="org.jboss.soa.esb.actions.bpel.BPELInvoke">
  <property name="service" value="
{http://www.jboss.org/bpel/examples/wsd1}HelloService"/>
  <property name="port" value="HelloPort" />
  <property name="operation" value="hello" />
  <property name="requestPartName" value="TestPart" />
  <property name="responsePartName" value="TestPart" />
</action>
```

ESB アクションクラスは、`org.jboss.soa.esb.actions.bpel.BPELInvoke` です。

このアクションのプロパティは以下のとおりです。

表6.1 プロパティ

プロパティ	説明
service	必須プロパティ。これは、デプロイされた BPEL プロセスと関連付けられた WSDL に登録されるサービス名を定義します。
port	任意のプロパティ。WSDL に登録するデプロイ済みの BPEL プロセスに関連付けられたポート名を定義します。このパラメーターは、ポート固有のエンドポイント設定情報が、BPEL プロセスデプロイメントの一部として登録されている場合のみ必要です。

プロパティ	説明
operation	必須プロパティ。呼び出される WSDL 操作を表します。
requestPartName	この任意プロパティを使い、ESB 受信メッセージのマッピング先となる WSDL メッセージの部分を定義します。ESB メッセージがすでにマルチパートメッセージを表現していない場合のみ、今プロパティを使用します。
responsePartName	この任意のプロパティを使用して、WSDL 応答メッセージのコンテンツを抽出し、パイプラインの次のアクションに渡される ESB メッセージに設定します (このプロパティが定義されていない場合、完全なマルチパートメッセージの値が ESB メッセージに設定されます)。
abortOnFault	この任意のオプションを使用して、BPEL プロセスの呼び出し時に発生したフォルトをメッセージとして処理するか (このプロパティの値が <b>false</b> の場合)、ESB サービスを中断する例外として処理します。デフォルト値は <b>true</b> で、サービスを中断させます。

この ESB アクションは、受信メッセージに対応しています。これらのメッセージの内容は以下のいずれかとして定義する必要があります。

表6.2 プロパティ

アイテム	説明
DOM	メッセージのコンテンツが DOM ドキュメントまたは要素の場合、完全なマルチパートメッセージとして利用するか、 <b>requestPartName</b> プロパティを使って定義したメッセージ部のコンテンツとして利用できます。
Java String	<p>メッセージのコンテンツは XML ドキュメントの文字列表現である場合、<b>requestPartName</b> は任意です。指定されていない場合、ドキュメントはマルチパートメッセージを表現する必要があります。</p> <p>メッセージコンテンツが XML ドキュメントを表現していない文字列の場合、<b>requestPartName</b> を指定する必要があります。</p>

メッセージコンテンツがマルチパートメッセージの各パートを表現している場合、トップレベルの要素として定義する必要があります (どのような名前でもつけることができる)。その子要素は、各メッセージパートを表現するため、直後に置く必要があります。代わりに、子要素は、それぞれの子要素を持つようになります。この要素は、名前部分の値を表現します。

マルチパートメッセージの構造の記述例を以下に示しています。

#### 例6.2 サンプル例

```
<message>
  <TestPart>
    Hello World
  </TestPart>
</message>
```

トップの要素(メッセージタグ)は重要ではありません。次の要素は、パート名を表現します。ここでは、**TestPart** というパートのみとなっています。この部分はテキストノードで、ここでは**Hello World**となっています(しかし、より複雑な XML 値の root ノードを表現する要素の場合もあります)。

### 6.2.1. フォルト処理

WSDL オペレーションが実行されている場合、応答メッセージは通常、BPELInvoke アクションから返されます。このメッセージは、アクションパイプラインに置かれ、次のアクションによる処理が可能になります(これ以上アクションが定義されていない場合、サービスクライアントに返されます)。

フォルトは少し違う方法で処理します。システムの設定方法に従い、ESB メッセージとしてフォルトを受信するか、アクションパイプラインを中断する例外を発生させることができます。どの動作を使用すべきか決定する設定プロパティは **abortOnFault** と呼ばれています。このプロパティのデフォルト値は、**true** です。**Loan Fault quick-start** を使ったサンプルを以下に示しています。

#### 例6.3 サンプル例

```
<action name="action2"
class="org.jboss.soa.esb.actions.bpel.BPELInvoke">
  <property name="service" value="{http://example.com/loan-
approval/wsdl/}loanService"/>
  <property name="operation" value="request" />
  <property name="abortOnFault" value="true" />
</action>
```

WSDL フォルトは、フォルトの詳細とタイプといった 2 種の情報をレポートします。それぞれ、メッセージボディの別部分で返されます。

#### 1. フォルトコード: `javax.xml.namespace.QName`

ESB メッセージボディの部分: `org.jboss.soa.esb.message.fault.detail.code`

このボディ部分は、BPEL プロセスにより返される固有の WSDL フォルトを指定します。



#### 警告

**QName used by the JBoss Enterprise SOA Platform** サーバーで使用する QName の固有バージョンは `lib/endorsed/stax-api.jar` ファイルに置かれています(このファイルがない場合、クラスバージョンの例外が発生します)。

#### 2. フォルトコード (QName のテキスト表現)

ESB メッセージボディの部分: `org.jboss.soa.bpel.message.fault.detail.code`

このボディの部分はフォルトコードの `QName` のテキスト表現を返します。このテキスト表現は、`{namespace}localpart` という形式をとります (`QName` に変換し直す場合は、`javax.xml.namespace.QName.valueOf(String)` メソッドを使います)。

### 3. フォルトの詳細

ESB メッセージボディの部分: `org.jboss.soa.esb.message.fault.detail.detail`

このボディの部分には、フォルトと関連付いているメッセージコンテンツのテキスト表現が含まれています。

## 6.2.2. SAML サポート

ESB サービスは *PicketLink* を使って *SAML token* を取得する場合、このアサーションは、呼び出された BPEL プロセスに `requestSAMLPartName` プロパティで渡すことができます。

### 例6.4 サンプル例

```
<action name="action2"
class="org.jboss.soa.esb.actions.bpel.BPELInvoke">
  <property name="service" value="
{http://simple_invoke/helloworld}HelloHeaderWSService"/>
  <property name="operation" value="sayHi" />
  <property name="requestPartName" value="sayHello" />
  <property name="responsePartName" value="sayHelloResponse" />
  <property name="requestSAMLPartName" value="Security" />
</action>
```

`requestSAMLPartName` は、メッセージ部の名前を指定します。WS-Security 要素としてこの部分を定義する必要があります。

### 例6.5 サンプル例

```
<part name="Security" element="wsse:Security"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" />
```

## 第7章 クラスタリングサポート

BPEL Console をクラスター環境で正しく機能させるには、クラスター環境のノード全体で共有されるデータベースを使うよう設定する必要があります (BPEL はデータベース内の処理ステータスをすべて永続化するためです)。

SOAP メッセージをノードに正しくディスパッチするには、ロードバランサーも必要です。

### 7.1. インストール

**riftsaw-clustering** ライブラリとファイルを **JBoss AS** にデプロイしたい場合は、**\$riftsaw/install** ファイルを開き、**deployment.properties** の下に **org\_jboss\_as\_config=all** を設定します。

または、別の設定でクラスタリング機能をデプロイしたい場合は、以下のコマンドを実行します。

```
ant deploy -Dclustering.support=true
```



#### 重要

クラスタリングサポートは、**JBoss AS Clustering** の **HAPartitionService** により左右されるため、このサービスが先に開始されていることを必ず確認してください。

### 7.2. 設定

以下の手順に従い BPEL クラスタリングを設定します。

1. **jboss-beans.xml** のサンプルファイルの **@database@** を以下のいずれかに設定します。
  - o mysql
  - o postgre
  - o db2
  - o sqlserver
  - o oracle
  - o sybase
2. **jboss-beans.xml** ファイルを **[server-config]/deploy/riftsaw.sar/META-INF/** にコピーします。
3. **[server-config]/deploy/cluster/jboss-cache-manager.sar/META-INF/jboss-cache-manager-jboss-beans.xml** を **riftsaw-cache-manager-jboss-beans.xml** に置き換えます。



**警告**

別の BPEL エンジンデプロイメントをインストールしようとする、統合が機能しなくなる可能性があります。

### 7.3. BPEL プロセスサービスの呼び出し

クラスターにデプロイしたサービスを呼び出したい場合、WSDL ファイルに SOAP アドレスではなく、ロードバランサーの URL を指定します。

## 付録A GNU GENERAL PUBLIC LICENSE 2.0

### GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but

does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange;  
or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that

system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED

TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along

with this program; if not, write to the Free Software Foundation, Inc.,

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Lesser General  
Public License instead of this License.



## 付録B 改訂履歴

改訂 5.2.0-0.1.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
改訂 5.2.0-0.1 翻訳完了	Thu Dec 6 2012	Credit Translator's
改訂 5.2.0-0.1 翻訳ファイルを XML ソース 5.2.0-0 と同期	Mon Dec 3 2012	Credit Translator's
改訂 5.2.0-0 初版	Thu May 30 2011	David Le Sage