



## WildFly スクリプト

### WildFly スタンドアロン アプリケーション サーバのスク リプト

スタンドアロンの WildFly の展開には、同じアプリケーション サーバまたは異なるサーバにインストールできる Service Link サーバと Service Catalog サーバが含まれます。

次に、スタンドアロン WildFly 展開用のインストーラで使用可能な起動および停止スクリプトを示します。

表 B-1 WildFly スタンドアロン アプリケーション サーバのスク립ト

スクリプト	実行場所	説明
startServiceCatalog	VM1	VM1 で Service Catalog を起動します。
startServiceLink	VM1	VM1 で Service Link を起動します。
stopServiceCatalog	VM1	VM1 で Service Catalog を停止します。
stopServiceLink	VM1	VM1 で Service Link を停止します。

### WildFly クラスタ アプリケーション サーバのスク リプト

WildFly のクラスタ セットアップは、2 VM トポロジまたは 4 VM トポロジで設定できます。

2 VM トポロジ :

- VM1 はホスト 1 のドメイン コントローラ、ホスト コントローラ、および Service Link です。
- VM2 はホスト 2 のホスト コントローラです。

4 VM トポロジ :

- VM1 はドメイン コントローラです。
- VM2 はスタンドアロン Service Link です。
- VM3 はホスト 1 のホスト コントローラです。
- VM4 はホスト 2 のホスト コントローラです。

表 B-2 WildFly クラスタ アプリケーション サーバのスクリプト

スクリプト	実行場所	説明
<b>2 VM クラスタ トポロジ :</b>		
<b>VM1 はドメイン コントローラ、HC1、および Service Link、VM2 は HC2 です。</b>		
startServiceCatalogCluster	VM1	VM1 で a) ドメイン コントローラ、b) プロセス コントローラ、c) Service Link の管理対象サーバ、および d) Service Catalog の管理対象サーバを起動します。   <b>(注)</b> startServiceCatalogCluster スクリプトは、展開された \$JBOSS_HOME/content/RequestCenter.war と \$JBOSS_HOME/content/ISEE.war がすでに使用可能かどうかを確認するために使用されます。使用可能であれば、スクリプトはコンテンツ フォルダを作成しません。使用可能でない場合は、WAR ファイルが /dist フォルダから展開され、それらのディレクトリの下にコンテンツ フォルダが作成されます。これは、2 VM トポロジでは VM1 と VM2 に適用され、4 VM トポロジでは VM3 と VM4 に適用されます。
startServiceCatalogOnHC1	VM1	VM1 で Service Catalog の管理対象サーバを起動します。
startServiceLinkOnHC1	VM1	VM1 で Service Link の管理対象サーバを起動します。
startServiceCatalogOnHC2	VM1	VM2 で Service Catalog の管理対象サーバを起動します。
startServiceCatalogOnHC3	VM1	VM3 で Service Catalog の管理対象サーバを起動します (ユーザがホスト コントローラ 3 を追加する場合のために予約済み)。
startServiceCatalogOnHC4	VM1	VM3 で Service Catalog の管理対象サーバを起動します (ユーザがホスト コントローラ 4 を追加する場合のために予約済み)。
deployServiceCatalogCluster all	VM1	a) ISEE.war および b) RequestCenter.war を展開します。
deployServiceCatalogCluster RC	VM1	RequestCenter.war を展開します。
deployServiceCatalogCluster SL	VM1	ISEE.war を展開します。
shutdownAllOnHC1	VM1	VM1 で a) ドメイン コントローラ、b) プロセス コントローラ、c) Service Link の管理対象サーバ、および d) Service Catalog の管理対象サーバを停止します。
forceStopAllOnHC	VM1	何らかの理由で shutdownAllOnHC1 スクリプトが機能しない場合は、このスクリプトを実行して VM1 で a) ドメイン コントローラ、b) プロセス コントローラ、c) Service Link の管理対象サーバ、および d) Service Catalog の管理対象サーバを強制的に停止します。
stopServiceCatalogOnHC1	VM1	VM1 で Service Catalog の管理対象サーバを停止します。
stopServiceLinkOnHC1	VM1	VM1 で Service Link の管理対象サーバを停止します。

表 B-2 (続き) WildFly クラスタ アプリケーション サーバのスクリプト

スクリプト	実行場所	説明
stopServiceCatalogOnHC2	VM1	VM2 で Service Catalog の管理対象サーバを停止します。
shutdownAllOnHC2	VM1	VM2 でプロセス コントローラと管理対象サーバの両方を停止します。
stopServiceCatalogOnHC3	VM1	VM3 で Service Catalog の管理対象サーバを停止します (ユーザが HC3 を追加する場合のために予約済み)。
shutdownAllOnHC3	VM1	VM3 でプロセス コントローラと管理対象サーバの両方を停止します (ユーザが HC3 を追加する場合のために予約済み)。
stopServiceCatalogOnHC4	VM1	VM3 で Service Catalog の管理対象サーバを停止します (ユーザが HC4 を追加する場合のために予約済み)。
shutdownAllOnHC4	VM1	VM4 でプロセス コントローラと管理対象サーバの両方を停止します (ユーザが HC4 を追加する場合のために予約済み)。
undeployServiceCatalogCluster all	VM1	a) ISEE.war および b) RequestCenter.war の展開を解除します。
undeployServiceCatalogCluster RC	VM1	RequestCenter.war の展開を解除します。
undeployServiceCatalogCluster SL	VM1	ISEE.war の展開を解除します。
startServiceCatalogCluster	VM2	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを起動します。 このスクリプトは、展開された \$JBOSS_HOME/content/RequestCenter.war と \$JBOSS_HOME/content/ISEE.war がすでに使用可能かどうかを確認するために使用されます。使用可能であれば、スクリプトはコンテンツ フォルダを作成しません。使用可能でない場合は、WAR ファイルが /dist フォルダから展開され、それらのディレクトリの下にコンテンツ フォルダが作成されます。
forceStopAllOnHC	VM2	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを強制的に停止します。
<b>お客様が別の VM を HC3 として追加する場合</b>		
startServiceCatalogCluster	VM3	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを起動します。
forceStopAllOnHC	VM3	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを強制的に停止します。
<b>お客様が別の VM を HC4 として追加する場合</b>		
startServiceCatalogCluster	VM4	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを起動します。
forceStopAllOnHC	VM4	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを強制的に停止します。
<b>4 VM クラスタ トポロジ :</b>		
<b>VM1 はドメイン コントローラ、VM2 は Service Link のスタンドアロン、VM3 は HC1、VM4 は HC2 です。</b>		
startServiceCatalogCluster	VM1	VM1 でドメイン コントローラを起動します。

## WildFly クラスタ アプリケーション サーバのスクリプト

表 B-2 (続き) WildFly クラスタ アプリケーション サーバのスクリプト

スクリプト	実行場所	説明
startServiceCatalogOnHC1	VM1	VM3 で Service Catalog の管理対象サーバを起動します。
startServiceCatalogOnHC2	VM1	VM4 で Service Catalog の管理対象サーバを起動します。
startServiceCatalogOnHC3	VM1	VM5 で Service Catalog の管理対象サーバを起動します (ユーザが HC3 を追加する場合のために予約済み)。
startServiceCatalogOnHC4	VM1	VM6 で Service Catalog の管理対象サーバを起動します (ユーザが HC4 を追加する場合のために予約済み)。
deployServiceCatalogCluster RC	VM1	RequestCenter.war を展開します。
stopDomainController	VM1	VM1 でドメイン コントローラを停止します。
stopServiceCatalogOnHC1	VM1	VM3 で Service Catalog の管理対象サーバを停止します。
shutdownAllOnHC1	VM1	VM3 でプロセス コントローラと管理対象サーバの両方を停止します。
stopServiceCatalogOnHC2	VM1	VM4 で Service Catalog の管理対象サーバを停止します。
shutdownAllOnHC2	VM1	VM4 でプロセス コントローラと管理対象サーバの両方を停止します。
stopServiceCatalogOnHC3	VM1	VM5 で Service Catalog の管理対象サーバを停止します (ユーザが HC3 を追加する場合のために予約済み)。
shutdownAllOnHC3	VM1	VM5 でプロセス コントローラと管理対象サーバの両方を停止します (ユーザが HC3 を追加する場合のために予約済み)。
stopServiceCatalogOnHC4	VM1	VM6 で Service Catalog の管理対象サーバを停止します (ユーザが HC4 を追加する場合のために予約済み)。
shutdownAllOnHC4	VM1	VM6 でプロセス コントローラと管理対象サーバの両方を停止します (ユーザが HC4 を追加する場合のために予約済み)。
undeployServiceCatalogCluster RC	VM1	RequestCenter.war の展開を解除します。
<b>VM1 はドメイン コントローラ、VM2 は Service Link のスタンドアロン、VM3 は HC1、VM4 は HC2 です。</b>		
StartServiceLink	VM2	VM2 で Service Link を起動します。
StopServiceLink	VM2	VM2 で Service Link を停止します。
<b>VM1 はドメイン コントローラ、VM2 は Service Link のスタンドアロン、VM3 は HC1、VM4 は HC2 です。</b>		
startServiceCatalogCluster	VM4	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを起動します。  このスクリプトは、展開された \$JBOSS_HOME/content/RequestCenter.war と \$JBOSS_HOME/content/ISEE.war がすでに使用可能かどうかを確認するために使用されます。使用可能であれば、スクリプトはコンテンツ フォルダを作成しません。使用可能でない場合は、WAR ファイルが /dist フォルダから展開され、それらのディレクトリの下にコンテンツ フォルダが作成されます。
forceStopAllOnHC	VM4	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを強制的に停止します。
<b>お客様が別の VM を HC3 として追加する場合</b>		
startServiceCatalogCluster	VM5	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを起動します。

表 B-2 (続き) WildFly クラスタ アプリケーション サーバのスクリプト

スクリプト	実行場所	説明
forceStopAllOnHC	VM5	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを強制的に停止します。
お客様が別の VM を HC4 として追加する場合		
startServiceCatalogCluster	VM6	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを起動します。
forceStopAllOnHC	VM6	a) プロセス コントローラおよび b) Service Catalog の管理対象サーバを強制的に停止します。

## 後続のホスト ノード (VM5) の手動での WildFly クラスタへの追加

### はじめる前に

1. 後続のノード (VM5) で設定されたホスト コントローラに対して GUI インストーラを実行します。
2. クラスタ ノードとして Host1 を選択した後、[ ノード タイプの選択 (Node Type Selection) ] パネルで Host Controller を選択します。
3. 起動スクリプトは実行しないでください。

**ステップ : 1** host1.xml の名前を host5.xml に変更します。host1.xml ファイルは、`InstallationDirectory/wildfly-8.2.0.Final/domain/configuration` ディレクトリの下にあります。

**ステップ : 2** 次の手順に従って新規ユーザを作成します。

- a. **add-user.sh** または **add-user.bat** スクリプトを `InstallationDirectory/wildfly-8.2.0.Final/bin` から実行します。
- b. 管理ユーザを選択するには、a と入力します。
- c. ユーザ名として **HOST5** を入力します。
- d. パスワードとして **HOST5** を入力します。
- e. 入力されたパスワードを使用するには、**yes** と入力します。
- f. パスワードの確認用に **HOST5** を再入力します。
- g. **Enter** を押して管理グループの情報を渡します。
- h. **Yes** と入力してユーザ「HOST5」を「ManagementRealm」領域に追加します。
- i. AS プロセスの相互接続には **no** と入力します。



(注)

- ホスト コントローラ (host5) およびドメイン コントローラ (host1) を持つマシンにも新規ユーザを作成する必要があります。
- ユーザが `InstallationDirectory/wildfly-8.2.0.Final/domain/configuration/mgmt-users.properties` に追加されていることを確認します。

**ステップ : 3** host5.xml で次の変更を実行します。

- a. ホストの名前を HOST1 から host5.xml の HOST5.location に変更します。このファイルは C:\Installation\_directory\wildfly-8.2.0.Final\domain\configuration にあります。

```
<host name="HOST5" xmlns="urn:jboss:domain:2.2">
```

- b. <server-identities> セクションのシークレット値を変更します。シークレット値は Web サイト (www.motobit.com) から入手できます。

```
<secret value="SE9TVDU=""/>
```

URL : http://www.motobit.com/util/base64-decoder-encoder.asp を入力します。

- テキスト ボックスに HOST5 と入力します (余分なスペースまたは改行なし)。
- [ ソース データを変換する (Convert the Source Data) ] ボタンをクリックします。

- c. 名前の競合を避けるため、サーバ インスタンスの名前を server-host1-RC から server-host5-RC に変更します。

```
<server name="server-host5-RC" group="main-server-group" auto-start="true">
```

- d. InstallationDirectory/bin ディレクトリで、setEnv.cmd スクリプトの CONTROLLER\_TYPE を host1 から host5 に変更します。

- e. InstallationDirectory/wildfly-8.2.0.Final/domain からログおよびサーバ ディレクトリを削除します (存在する場合)。

**ステップ : 4** 以下の変更を InstallationDirectory/bin ディレクトリの startServiceCatalogCluster.cmd スクリプトで実行する必要があります。

- a. スクリプト ファイルに以下のコード スニペットを太字で追加します。

```
#Below if condition is applicable for the windows OS platform

if "%CONTROLLER_TYPE%"=="host5" (
    if exist "%WILDFLY_BASE_DIR%\configuration\domain.xml" rename
    "%WILDFLY_BASE_DIR%\configuration\domain.xml" "domain_backup.xml"
)

#Below if condition is applicable for the Linux OS platform

#pause 'Press [Enter] key to continue6...'
FILE=${WILDFLY_BASE_DIR}/configuration/host5.xml
if [ "${CONTROLLER_TYPE}" == "host5" ]
次に
    if [ -f "$FILE" ]
        次に
            /bin/mv -i ${WILDFLY_BASE_DIR}/configuration/host5.xml
fi
    fi
fi
```

- b. スクリプト ファイルに以下のコード スニペットを太字で追加します。

```
if "%CONTROLLER_TYPE%" == "host5" if exist "%WILDFLY_BASE_DIR%\configuration\host5.xml"
rename "%WILDFLY_BASE_DIR%\configuration\host5.xml" "host.xml"
```

If the Platform is Linux OS , skip the step b and proceed. above if condition only applicable for the windows OS platform.

- ステップ : 5** startServiceCatalogCluster.cmd または sh を使用してサーバを起動し、立ち上がったら host5 が登録されているかどうかをこのドメインの下の Wildfly 管理サーバ コンソールから確認します。
- ステップ : 6** VM-1 マシン (ドメイン コントローラを含むマシン) での 4-VM クラスタおよび 2-VM クラスタ両方のスクリプトの追加
- a. shutdownAllOnHC1.cmd をコピーし、shutdownAllOnHC5.cmd としてコピー済みスクリプトの名前を変更します
  - b. startServiceCatalogOnHC1.cmd をコピーし、startServiceCatalogOnHC5.cmd としてコピー済みスクリプトの名前を変更します
  - c. stopServiceCatalogOnHC1.cmd をコピーし、stopServiceCatalogOnHC5.cmd としてコピー済みスクリプトの名前を変更します



**(注)** Linux プラットフォームでは、新しいスクリプトに読み取り / 書き込み権限がない場合、実行前に手動で読み取り / 書き込み権限を割り当てる必要があります。

- d. stopServiceCatalogOnHC5.cmd スクリプト ファイルで次の操作を慎重に実行します。
  - HOST1 の名前を HOST5 に変更します。
  - host1 の名前を host5 に変更します。

## IIS Web サーバのプラグインの設定

このセクションでは、Windows Server 2012 R2 マシンで IIS Web サーバ (バージョン 8.x) 用のプラグインを設定し、Wildfly クラスタ アプリケーション サーバ (バージョン 8.2.0.Final) にリダイレクトする方法を説明します。

IIS Web サーバのプラグインを設定するには :

- ステップ : 1** [IIS の Web サーバ の役割の追加](#)
- ステップ : 2** [Tomcat プラグインのインストール](#)
- ステップ : 3** [WAR ディレクトリのコピー](#)
- ステップ : 4** [IIS の仮想ディレクトリの作成](#)
- ステップ : 5** [プラグイン プロパティの変更](#)
- ステップ : 6** [WildFly インスタンス ID の設定](#)
- ステップ : 7** [IIS のテスト](#)

## IIS の Web サーバの役割の追加

- 
- ステップ : 1 [ サービス マネージャ ダッシュボード (Service Manager Dashboard) ] > [ 管理タブ (Manage Tab) ] > [ 役割と機能の追加 (Add Roles and Features) ] に移動します。
  - ステップ : 2 Windows Server 2012 R2 オペレーティング システムで「Web サーバ (IIS)」の役割を追加します。IIS を追加するとき、必ず「ISAPI 拡張」および「ISAPI フィルター」の役割のサービスを選択してください。
  - ステップ : 3 「Web サーバ (IIS)」の役割を追加したら、World Wide Web Publishing Service を開始します。
  - ステップ : 4 ブラウザを起動し、URL = `http://localhost` に接続します。  
[ IIS 8.x へようこそ (Welcome IIS 8.x) ] ページが表示されます。

## Tomcat プラグインのインストール

- 
- ステップ : 1 cisco.com から PSC 11.x ソフトウェア パッケージをダウンロードし、展開します。
  - ステップ : 2 <PSC\_Software\_Extract\_Dir>\isapi ディレクトリに移動します。
  - ステップ : 3 ファイル `tomcat-isapi-redirector-v1.2.37.zip` をお使いの IIS Web サーバ マシンにコピーし、`C:\inetpub\isapi` ディレクトリの下に展開します。



(注) 必要に応じて、最初に `isapi` サブディレクトリを `C:\inetpub` の下に作成します。

- 
- ステップ : 4 `C:\inetpub\isapi` に移動し、`isapi_redirect-1.2.37-win64.dll` ファイルの名前を `isapi_redirect.dll` に変更します。

## WAR ディレクトリのコピー

WildFly をインストールしていて、IIS が WildFly とは別のマシン上にある場合、WAR ディレクトリをコピーするには次のステップを実行します。

### WildFly のスタンドアロン インストール

- 
- ステップ : 1 RequestCenter Server が実行されている WildFly マシン上で、<PSC\_Install\_Dir>\wildfly-8.2.0.Final\ServiceCatalogServer\deployments ディレクトリに移動します。
  - ステップ : 2 このマシンから `RequestCenter.war` サブディレクトリ全体を IIS マシンにコピーし、`C:\inetpub\WAR\` ディレクトリの下に配置します。
  - ステップ : 3 ServiceLink Server が実行されている WildFly マシン上で、<PSC\_Install\_Dir>\wildfly-8.2.0.Final\ServiceLinkServer\deployments ディレクトリに移動します。
  - ステップ : 4 このマシンから `ServiceLink.war` サブディレクトリ全体を IIS マシンにコピーし、`C:\inetpub\WAR\` ディレクトリの下に配置します。



#### 4 VM トポロジの WildFly クラスタ インストール

- 
- ステップ : 1** RequestCenter のクラスタ ノードのいずれかが実行されている WildFly マシン上で、<PSC\_Install\_Dir>\wildfly-8.2.0.Final\content ディレクトリに**移動**します。
- ステップ : 2** このマシンから RequestCenter.war サブディレクトリ全体を IIS マシンにコピーし、C:\inetpub\WAR\ ディレクトリの下に配置します。
- ステップ : 3** ServiceLink Server が実行されている WildFly マシン上で、<PSC\_Install\_Dir>\wildfly-8.2.0.Final\ServiceLinkServer\deployments ディレクトリに**移動**します。
- ステップ : 4** このマシンから ServiceLink.war サブディレクトリ全体を IIS マシンにコピーし、C:\inetpub\WAR\ ディレクトリの下に配置します。

#### 2 VM トポロジの WildFly クラスタ インストール

- 
- ステップ : 1** RequestCenter のプライマリ クラスタ ノードが実行されている WildFly マシン上で、<PSC\_Install\_Dir>\wildfly-8.2.0.Final\content ディレクトリに**移動**します。
- ステップ : 2** このマシンから RequestCenter.war サブディレクトリ全体を IIS マシンにコピーし、C:\inetpub\WAR\ ディレクトリの下に配置します。
- ステップ : 3** 同じ WildFly マシン上で、<PSC\_Install\_Dir>\wildfly-8.2.0.Final\content ディレクトリに**移動**します。
- ステップ : 4** このマシンから ISEE.war サブディレクトリ全体を IIS マシンにコピーし、C:\inetpub\WAR ディレクトリの下に配置します。
- ステップ : 5** フォルダ名を C:\inetpub\WAR\ServiceLink.war に変更します。

## IIS の仮想ディレクトリの作成

- 
- ステップ : 1** IIS マシン上で、**Internet Information Services (IIS) マネージャ**を起動します。
- ステップ : 2** [ ホスト名 (Hostname) ] > [ サイト (Sites) ] > [ デフォルト Web サイト (Default Web Site) ] を選択します。
- ステップ : 3** [ デフォルト Web サイト (Default Web Site) ] を右クリックし、[ 仮想ディレクトリを追加 (Add Virtual Directory) ] を選択します。
- ステップ : 4** ポップアップ ウィンドウで次の値を入力し、[OK] をクリックします。  
Alias = RequestCenter  
Physical path = <Click the browse button, and select the "C:\inetpub\WAR\RequestCenter.war" directory.>
- ステップ : 5** [ デフォルト Web サイト (Default Web Site) ] を右クリックし、[ 仮想ディレクトリを追加 (Add Virtual Directory) ] を選択して別のディレクトリを追加します。
- ステップ : 6** ポップアップ ウィンドウで次の値を入力し、[OK] をクリックします。  
Alias = IntegrationServer  
Physical path = <Click the browse button, and select the "C:\inetpub\WAR\ServiceLink.war" directory.>
- ステップ : 7** [ デフォルト Web サイト (Default Web Site) ] を右クリックし、[ 仮想ディレクトリを追加 (Add Virtual Directory) ] を選択して別のディレクトリを追加します。

**ステップ : 8** ディスプレイ ウィンドウで次の値を入力し、[OK] をクリックします。

```
Alias = tomcat
Physical path = C:\inetpub\isapi
```

**ステップ : 9** [ デフォルト Web サイト (Default Web Site) ] ノードをクリックします。右ペインで、[ISAPI フィルタ (ISAPI Filters) ] をダブルクリックします。

**ステップ : 10** 右ペインの [ アクション (Actions) ] 列の下にある [ 追加 (Add) ] リンクをクリックします。

**ステップ : 11** ポップアップ ウィンドウで次の値を入力し、[OK] をクリックします。

```
Filter name = tomcat
Executable = C:\inetpub\isapi\isapi_redirect.dll
```

**ステップ : 12** [ デフォルト Web サイト (Default Web Site) ] ノードをクリックします。右ペインで、[ ハンドラ マッピング (Handler Mappings) ] をダブルクリックします。

**ステップ : 13** 右ペインの [ アクション (Actions) ] 列の下にある [ 機能権限の編集 (Edit Feature Permissions) ] リンクをクリックします。

**ステップ : 14** ポップアップ ウィンドウで、[ すべて読み込み (all Read) ]、[ スクリプト (Script) ] および [ 実行 (Execute) ] チェックボックスを選択し、[OK] をクリックします。

**ステップ : 15** [ ホスト名 (Hostname) ] ノードをクリックします。右側のパネルで、[ISAPI と CGI の制限 (ISAPI and CGI Restrictions) ] をダブルクリックします。




---

(注) [ ホスト名 (Hostname) ] ノードはサイトの親ノードです。

---

**ステップ : 16** 最も右側にあるペインの [ アクション (Actions) ] 列の下にある [ 追加 (Add) ] リンクをクリックします。

**ステップ : 17** ディスプレイ ウィンドウで次の値を入力し、[OK] をクリックします。

```
ISAPI or CGI path = C:\inetpub\isapi\isapi_redirect.dll
Description = Tomcat ISAPI Filter
Select the checkbox "Allow extension path to execute".
```

## プラグイン プロパティの変更

**ステップ : 1** ファイル `C:\inetpub\isapi\isapi_redirect.properties` を以下のように変更します。

```
# Configuration file for the Jakarta ISAPI Redirector
# The path to the ISAPI Redirector Extension, relative to the website
# This must be in a virtual directory with execute privileges
extension_uri=/tomcat/isapi_redirect.dll
# Full path to the log file for the ISAPI Redirector
log_file=C:\inetpub\isapi\logs\isapi_redirect.log
# Log level (debug, info, warn, error or trace)
log_level=error
# Full path to the workers.properties file
worker_file=C:\inetpub\isapi\conf\workers.properties
# Full path to the uriworkermap.properties file
worker_mount_file=C:\inetpub\isapi\conf\uriworkermap.properties
```

**ステップ : 2** ファイル `C:\inetpub\isapi\conf\uriworkermap.properties` を以下のように変更します。

```
/RequestCenter=router1
/RequestCenter/*=router1
/RequestCenter/servlet/*=router1
```

```

/IntegrationServer=router2
/IntegrationServer/*=router2
/IntegrationServer/servlet/*=router2

/private/admin/jkstatus=jkstatus

```

**ステップ : 3** WildFly のスタンドアロン インストールがあれば、ファイル **C:\inetpub\isapi\conf\workers.properties** を以下のように変更します。

```

# Define list of workers that will be used for mapping requests
worker.list=router1,router2,jkstatus

# Define Node1 worker for RequestCenter
worker.node1.port=8009
worker.node1.host=<IP_Address_of_RC_host>
worker.node1.type=ajp13
worker.node1.lbfactor=1
worker.router1.type=lb
worker.router1.balance_workers=node1

# Define Node2 worker for ServiceLink
worker.node2.port=6009
worker.node2.host=<IP_Address_of_SL_host>
worker.node2.type=ajp13
worker.node2.lbfactor=1
worker.router2.type=lb
worker.router2.balance_workers=node2

# Define a jkstatus worker using status
worker.jkstatus.port=8009
worker.jkstatus.host=<IP_Address_of_IIS_host>
worker.jkstatus.type=status
worker.status.type=status

```

ポート番号 8009 と 6009 は、*RequestCenter server* と *ServiceLink server* が使用する実際の *ajp* ポート番号に設定してください。

**ステップ : 4** WildFly クラスタ インストールがあれば (4-VM トポロジか 2-VM トポロジかを問わず)、ファイル **C:\inetpub\isapi\conf\workers.properties** を以下のように変更します。



(注) 以下は、*RequestCenter* の 2 つのクラスタ ノードを持つ WildFly のインストールの例を示します。

- つまり、2 つの個別 VM 上で 2 つの *RequestCenter* サーバが動作している状況です。
- 3 つまたは 4 つのクラスタ ノードがある場合は、この例に従って、*rcnode3* と *rcnode4* のセクションを適宜追加するだけです。

```

# Define list of workers that will be used for mapping requests
worker.list=router1,router2,jkstatus

# Define rcnode1 worker for RequestCenter 1
worker.rcnode1.port=8009
worker.rcnode1.host=<IP_Address_of_RC1_host>
worker.rcnode1.type=ajp13
worker.rcnode1.lbfactor=1

# Define rcnode2 worker for RequestCenter 2
worker.rcnode2.port=8009
worker.rcnode2.host=<IP_Address_of_RC2_host>
worker.rcnode2.type=ajp13
worker.rcnode2.lbfactor=1

# For clustering, set the line below to rcnode1, rcnode2, etc...

```

```

worker.router1.type=lb
worker.router1.balance_workers=rcnode1,rcnode2

# Define slnode worker for ServiceLink
worker.slnode.port=6009
worker.slnode.host=<IP_Address_of_SL_host>
worker.slnode.type=ajp13
worker.slnode.lbfactor=1

worker.router2.type=lb
worker.router2.balance_workers=slnode

# Define a 'jkstatus' worker using status
worker.jkstatus.port=8009
worker.jkstatus.host=<IP_Address_of_IIS_host>
worker.jkstatus.type=status
worker.status.type=status

```

**ステップ : 5** World Wide Web Publishing Service を再起動します。

**ステップ : 6** WildFly のスタンドアロン インストールがあれば、セクション [IIS のテスト](#) に進んでください。



(注) Wildfly のクラスタ インストールがある場合、次のセクション「Wildfly のインスタンス ID の設定」に進みます。

## WildFly インスタンス ID の設定

### 4 VM トポロジの場合

4 VM トポロジの WildFly クラスタ インストールでは、次の手順を実行する必要があります。

**ステップ : 1** WildFly ドメイン コントローラ マシンにログインし、WildFly のすべてのサーバを停止します。

**ステップ : 2** ファイル <PSC\_Install\_Dir>\wildfly-8.2.0.Final\domain\configuration\domain.xml を開き、以下の行を検索します。

```
<subsystem xmlns="urn:jboss:domain:undertow:1.2">
```

これを以下の値に置換します。

```
<subsystem xmlns="urn:jboss:domain:undertow:1.2" instance-id="${jboss.web.instanceId}">
```

**ステップ : 3** RequestCenter で Host Controller 1 にログインし、すべての WildFly サーバを停止します。

**ステップ : 4** ファイル <PSC\_Install\_Dir>\wildfly-8.2.0.Final\domain\configuration\host1\_backup.xml を開き、以下のセクションを検索します。

```

<servers>
<server name="server-host1-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<!--<option value="-Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscale/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>

```

```

<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
</server>
</servers>

```

これを以下に置換します。

```

<servers>
<server name="server-host1-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<!--<option value="-Xrunjdw:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscale/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
<system-properties>
  <property name="jboss.web.instanceId" value="rcnode1"/>
</system-properties>
</server>
</servers>

```

**ステップ : 5 RequestCenter で Host Controller 2 にログインし、すべての Wildfly サーバを停止します。**

**ステップ : 6** ファイル <PSC\_Install\_Dir>\wildfly-8.2.0.Final\domain\configuration\host2\_backup.xml を開き、以下のセクションを検索します。

```

<servers>
<server name="server-host2-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<!--<option value="-Xrunjdw:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscale/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
</server>
</servers>

```

これを以下に置換します。

```
<servers>
<server name="server-host2-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<!--<option value="-Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscale/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
<system-properties>
  <property name="jboss.web.instanceId" value="rcnode2"/>
</system-properties>
</server>
</servers>
```



(注) 複数のクラスタ ノードがあれば、その後の各ノードに対して [ステップ : 5](#) と [ステップ : 6](#) を繰り返します。次に例を示します。

- 3 番目のノードで、*host3\_backup.xml* ファイルを変更し、*rcnode3* のシステム プロパティを追加する必要があるとします。
- 4 番目のノードで、*host4\_backup.xml* ファイルを変更し、*rcnode4* のシステム プロパティを追加する必要があるとします。

**ステップ : 7** Wildfly サーバをドメイン コントローラ マシンと各ホスト コントローラ マシンで開始します。

**ステップ : 8** セクション「[IIS のテスト](#)」に進みます。

## 2 VM トポロジの場合

2 VM トポロジの WildFly クラスタ インストールがない限り、このセクションの手順は実行しないでください。

**ステップ : 1** WildFly ドメイン コントローラ マシンにログインし、WildFly のすべてのサーバを停止します。

**ステップ : 2** ファイル `<PSC_Install_Dir>\wildfly-8.2.0.Final\domain\configuration\domain.xml` を開き、以下の行を検索します。

```
<subsystem xmlns="urn:jboss:domain:undertow:1.2">
```

これを以下の値に置換します。

```
<subsystem xmlns="urn:jboss:domain:undertow:1.2" instance-id="{jboss.web.instanceId}">
```

**ステップ : 3** ファイル `<PSC_Install_Dir>\wildfly-8.2.0.Final\domain\configuration\hostva_backup.xml` を開き、以下のセクションを検索します。

```
<servers>
<server name="server-host1-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
```

```

<jvm-options>
<!--<option value="-Xrunjdw:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscaler/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
</server>
<server name="server-host1-SL" group="other-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<option value="-server"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="standard-sockets" port-offset="0"/>
</server>
</servers>

```

これを以下に置換します。

```

<servers>
<server name="server-host1-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<!--<option value="-Xrunjdw:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscaler/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
<system-properties>
  <property name="jboss.web.instanceId" value="rcnode1"/>
</system-properties>
</server>
<server name="server-host1-SL" group="other-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<option value="-server"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="standard-sockets" port-offset="0"/>
<system-properties>
  <property name="jboss.web.instanceId" value="slnode"/>
</system-properties>
</server>
</servers>

```

**ステップ : 4** RequestCenter で Host Controller 2 にログインし、すべての WildFly サーバを停止します。

**ステップ : 5** ファイル <PSC\_Install\_Dir>\wildfly-8.2.0.Final\domain\configuration\host2\_backup.xml を開き、以下のセクションを検索します。

```
<servers>
<server name="server-host2-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<!--<option value="-Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscale/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
</server>
</servers>
```

これを以下に置換します。

```
<servers>
<server name="server-host2-RC" group="main-server-group" auto-start="true">
<jvm name="default">
<heap size="2048m" max-size="2048m"/>
<permgen size="512m" max-size="512m"/>
<jvm-options>
<!--<option value="-Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n"/>-->
<option
value="-XX:CompileCommand=exclude,com/newscale/bfw/signon/filters,AuthenticationFilter"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xml/dtm/ref/sax2dtm/SAX2DTM,startElement"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option value="-XX:CompileCommand=exclude,org/exolab/castor/xml/Marshaller,marshal"/>
<option
value="-XX:CompileCommand=exclude,org/apache/xpath/compiler/XPathParser,UnionExpr"/>
</jvm-options>
</jvm>
<socket-bindings socket-binding-group="ha-sockets" port-offset="0"/>
<system-properties>
  <property name="jboss.web.instanceId" value="rcnode2"/>
</system-properties>
</server>
</servers>
```



**(注)** 複数のクラスタ ノードがあれば、その後の各ノードに対して **ステップ : 4** と **ステップ : 5** を繰り返します。次に例を示します。

- 3 番目のノードで、*host3\_backup.xml* ファイルを変更し、*rcnode3* のシステム プロパティを追加する必要があるとします。
- 4 番目のノードで、*host4\_backup.xml* ファイルを変更し、*rcnode4* のシステム プロパティを追加する必要があるとします。

**ステップ : 6** WildFly サーバをドメイン コントローラ マシンとホスト コントローラ 2 マシンで開始します。

**ステップ : 7** セクション「IIS のテスト」に進みます。



## IIS のテスト

以下のセクションでは、Prime Service Catalog への接続の確認に関する情報を提供します。

- 
- ステップ : 1** URL `http://<IP_Address_of_IIS_Host>/RequestCenter` に接続することで確認します。接続できている場合、Prime Service Catalog のログイン ページが表示されます。
  - ステップ : 2** 2 つのノードの WildFly クラスタ インストールでは、ノードの 1 つを停止します。URL `http://<IP_Address_of_IIS_Host>/RequestCenter` にまだ接続できることを確認します。
- 

## WildFly での SQL のデータソースの設定

- 
- ステップ : 1** クレデンシャルを使用して WildFly Admin コンソール (下記の URL の例) にログオンし、[OK] をクリックします。WildFly Application 管理コンソールが表示されます。  
URL の例 :  
`http://<hostname>:<port>/RequestCenter`  
値は次のとおりです。  
<hostname> : Service Catalog の WildFly サーバをインストールしたコンピュータの完全修飾ドメイン ホスト名  
または IP アドレス。  
<port> : Service Catalog の Wildfly サーバに割り当てられている HTTP ポート番号。HTTP ポート番号のデフォルト値は 8080 です。
  - ステップ : 2** 編集と変更を行うため、[ 設定 (Configuration) ] タブをクリックします。
  - ステップ : 3** 新しいデータソースを追加するため、[ 追加 (Add) ] をクリックします。
  - ステップ : 4** [ 名前 (Name) ] に **SERVICECATALOGDS** と入力し、[ JNDI 名 (JNDI Name) ] に **java:<データソース名>** と入力します。
  - ステップ : 5** [ 次へ (Next) ] をクリックします。
  - ステップ : 6** ドライバとして [Microsoft] を選択し、[ 次へ (Next) ] をクリックします。
  - ステップ : 7** 接続 URL として **jdbc:sqlserver://<db\_server>:1433;DatabaseName=VM236\_RCDB\_RC4** を入力します。
  - ステップ : 8** クレデンシャルを入力します。

**ステップ : 9** 次の表に示すように、各タブの各種フィールドに値を入力します。

タブ名	フィールド名	値
セキュリティ	[ セキュアドメイン (Secure Domain) ]	CiscoSecureDataSource
接続	[JTA を使用 (Use JTA) ] と [CCMsECURITY を使用 (Use CCMsECURITY) ]	両方のオプションをオンにする必要があります。
プロパティ	1. SelectMethod:  2. sendStringParametersAsUnicode:	1. [ 直接 (Direct) ]  2. [ はい (True) ]
プール	最小サイズは 20、最大サイズは 80、その他の値は False に設定します。	
検証	[ バックグラウンド検証 (Background Validation) ]	いいえ (False)
	[ 検証ミリ秒 (Validation Millis) ]	90000
	[ 一致時に検証 (Validate on Match) ]	いいえ (False)

**ステップ : 10** [ 有効化 (Enable) ] > [ 確認 (Confirm) ] をクリックします。



(注) (WildFly クラスタにのみ適用) データソースが依然として無効の場合は、domain.xml ファイルでデータソースを <enabled = true> に設定し、ドメインを再起動します。

**ステップ : 11** [ 接続 (Connection) ] タブで接続をテストすると、「JDBC 接続が正常に作成されました (Successfully created the JDBC connection.)」という確認メッセージが表示されます。

**ステップ : 12** WildFly サーバを再起動します。

## WildFly での Oracle のデータ ソースの設定

**ステップ : 1** クレデンシヤルを使用して WildFly Admin コンソール (下記の URL の例) にログオンし、[OK] をクリックします。編集と変更を行うため、[設定 (Configuration) ] タブをクリックします。

URL の例 :

<http://<hostname>:<port>/RequestCenter>

値は次のとおりです。

<hostname> : Service Catalog の WildFly サーバをインストールしたコンピュータの完全修飾ドメイン ホスト名

または IP アドレス。

<port> : Service Catalog の Wildfly サーバに割り当てられている HTTP ポート番号。HTTP ポート番号のデフォルト値は 8080 です。

- ステップ : 2** 新しいデータソースを追加するため、[ 追加 (Add) ] をクリックします。
- ステップ : 3** [ 名前 (Name) ] に **SERVICECATALOGDS** と入力し、[ JNDI 名 (JNDI Name) ] に **java:/<データソース名>** と入力します。
- ステップ : 4** [ 次へ (Next) ] をクリックします。
- ステップ : 5** ドライバとして [ Oracle シン (oracle-thin) ] を選択し、[ 次へ (Next) ] をクリックします。
- ステップ : 6** 接続 URL として `jdbc:oracle:thin:@//<db_server>:1433;DatabaseName=VM236_RCDB_RC4` を入力します。
- ステップ : 7** クレデンシャルを入力します。
- ステップ : 8** 次の表に示すように、各タブの各種フィールドに値を入力します。

タブ名	フィールド名	値
セキュリティ	[ セキュアドメイン (Secure Domain) ]	CiscoSecureDataSource
接続	[ JTA を使用 (Use JTA) ] と [ CCM を使用 (Use CCM) ]	両方のオプションをオンにする必要があります。
プロパティ	1. SelectMethod: 2. sendStringParametersAsUnicode:	1. [ 直接 (Direct) ] 2. [ はい (True) ]
プール	最小サイズは 20、最大サイズは 80、その他の値は False に設定します。	
検証	[ バックグラウンド検証 (Background Validation) ]	いいえ (False)
	[ 検証ミリ秒 (Validation Millis) ]	90000
	[ 一致時に検証 (Validate on Match) ]	いいえ (False)

- ステップ : 9** [ 有効化 (Enable) ] > [ 確認 (Confirm) ] をクリックします。



**(注)** (WildFly クラスタにのみ適用) データソースが依然として無効の場合は、`domain.xml` ファイルでデータソースを `<enabled = true>` に設定し、ドメインを再起動します。

- ステップ : 10** [ 接続 (Connection) ] タブで接続をテストすると、「JDBC 接続が正常に作成されました (Successfully created the JDBC connection.)」という確認メッセージが表示されます。

- ステップ : 11** WildFly サーバを再起動します。

