



# TIBCO® BPM Enterprise

## Deployment

*Version 4.3.2*

*May 2022*



# Contents

---

<b>Introduction to Application Deployment</b> .....	<b>4</b>
About Project Versions .....	4
Project Version Numbers .....	5
Version Usage Best Practices .....	5
How BPM Handles Organization Model Versions at Deployment .....	5
Amending Version Numbers .....	7
Changing the Major Version Number of an Organization Model .....	8
<b>Deploying BPM Applications</b> .....	<b>9</b>
Deployment from TIBCO Business Studio Direct Or DAA Export .....	9
Deploying a Process That Exposes a Web Service .....	10
Deploying a Global Signal Definition Project .....	10
Business Process Execution Language (BPEL) and Deployment .....	11
Preparation for Deployment .....	11
Creating a New Deployment Server .....	12
Local Development Server .....	17
Enabling Deployment Over Secured (https) Communications for a Deployment Server .....	17
Connecting to a Deployment Server .....	18
Refreshing a Deployment Server .....	20
Disconnecting From a Deployment Server .....	20
Viewing and Changing Deployment Server Properties .....	20
Accessing Runtime Clients and Administrator Server from TIBCO Business Studio .....	20
Deploying an Application from TIBCO Business Studio .....	21
Deploying an Application Using the Deployment Wizard .....	21
Deployment of Global Signal Definition Project and Process with Global Signal Events .....	24
Deploying an Application by Dragging and Dropping .....	24
Verifying the Deployment .....	24
Exporting to DAA .....	25
DAA Validation .....	26
Deploying an Application from Administrator .....	27
Deployed Applications .....	28
Deploying the Organization Model .....	28
Deploying a BPM Application Fails If the Referenced Organization Model Has Been Upgraded .....	29
Deployment to Multiple BPM Systems .....	29
Re-Deploying for Live Development a Project that Has Changed .....	30
Re-Deploy/Upgrade of a Global Signal Definition Run-time Application .....	31
<b>Scripted Deployment</b> .....	<b>32</b>

Creating a Deployment Project .....	32
Adding Resources to the Deployment Project .....	33
Specifying the Server .....	34
Configuring the Resources in the Deployment Project .....	35
Generating the Deployment Scripts .....	40
Modifying a Deployment Project .....	42
Adding a Resource .....	42
Removing a Resource .....	42
Reconfiguring a Resource .....	43
Changing the Deployment Order of Resources .....	43
Executing the Deployment Script .....	43
Checking in and out of a Repository .....	43
Deploying the Project .....	44
Sample Projects - Shared Resource Create Install and Subversion (SVN) .....	45
Creating Installing Shared Resources .....	45
Using Subversion .....	45
<b>Undeployment .....</b>	<b>46</b>
Undeploying an Application Directly from TIBCO Business Studio .....	46
Undeploying an Application Using Administrator .....	46
Undeploying an Application Using a Deployment Script .....	47
Tips for Successful Undeployment .....	48
Cleaning Up Undeployed Applications .....	48
<b>Application Upgrade .....</b>	<b>50</b>
Upgrading a Deployed Application .....	50
Reverting to the Original Version of an Upgraded Application .....	50
Upgrading an Application Using a Deployment Server .....	51
Upgrading an Application Using a Deployment Script .....	51
<b>Deploying Language Packs .....</b>	<b>53</b>
<b>TIBCO Documentation and Support Services .....</b>	<b>54</b>
<b>Legal and Third-Party Notices .....</b>	<b>56</b>

# Introduction to Application Deployment

---

This document tells you how to deploy and undeploy, on a TIBCO® BPM Enterprise (formerly TIBCO ActiveMatrix® BPM) installation, BPM applications that have been designed in TIBCO Business Studio. It also tells you how to go about upgrading deployed projects when new versions have been produced in TIBCO Business Studio.

An application is deployed to BPM by deploying one or more TIBCO Business Studio *projects* to a BPM server. The project is the unit of deployment, so you cannot choose to deploy only part of a project.

A project can contain one or more of the following:

- Packages. A package is a container, in XPD format, for one or more processes.
- Processes. A process models the actual business process in an organization. A process may include sub-processes.
- Forms. You can define forms to collect user input in a user task within a business process
- Business Object Model. A business object model is a set of business terms and relationships specific to your corporate environment (for example, in a financial environment, broker, counterparty, and so on).
- Emulations. You can add emulation files that allow you to emulate a process and add test data at certain activities, thus enabling you to check the data flow at various activities in a process.
- Organization Models. An organization model defines the organizational structure of your enterprise and the relationships between the different components (for example, organization units and positions) within your organization.
- Service Descriptors. Service assets include the WDSL files for any web services that you import into your project.
- TIBCO SOA Platform. If you want to deploy an application to the TIBCO SOA Platform, you need to create the special folders to contain the SOA assets. See in the appropriate BPM Composite Development Guide for more information about developing applications for use with the TIBCO SOA Platform.
- Business Assets. There are two categories of project-related business assets in TIBCO Business Studio:
  - Quality Process. Business cases, project plans, and so on.
  - Ad-hoc Assets. Supporting documents, spreadsheets, and so on that are not part of the quality process.

Not all of these need be in the same project: see "Distribution of Assets Across Multiple Projects" in the *TIBCO Business Studio BPM Implementation Guide*.

However the project is the unit of deployment, so you cannot deploy a subset of a project. For example, if you are designing an application that uses a main business process with sub-processes depending on it, you might put the sub-processes into different XPD packages within the same project, so that different people can easily work on them at the same time. Because the sub-processes are all part of the same project, if one sub-process changes you cannot redeploy it separately. If you have processes which you do want to deploy separately, you must organize them initially into separate projects.

## About Project Versions

There are two types of version numbers in TIBCO Business Studio, Project version numbers and XPD Package version numbers.

- Project version numbers. Organization models use project numbers to control the interaction between different organization models with applications in BPM. You should only need to update project version numbers if you want to make changes to your organization models and then redeploy them to BPM. See [Project Version Numbers](#).

- XPDL Package version numbers. Processes use XPDL package version numbers to control interaction between processes in different XPDL packages. You should only need to update XPDL version numbers if you want to make changes to a process in an individual XPDL package but do not want to have to redeploy other projects that reference other processes in other XPDL packages in the same project.

Note that:

- Apart from processes, all other project artifacts ignore XPDL version numbers.
- An XPDL version number overrides a project version number.

## Project Version Numbers

Project version numbers are used to control the interaction of different organization models with applications in BPM.

At design time, in TIBCO Business Studio:

- a BPM Developer project, Analysis project, or a project containing a business object model or an organization model, is given a version number, in the form *major.minor.micro.qualifier*. This defaults to *1.0.0.qualifier* when the project is created.
- a process definition, if it is not created in the same project as the organization model it uses, references the project containing that organization model.
- within a project, all references must be to the same **major version** of the organization model.



If you upgrade and change the major version number on a sub-process (for example, from 1.0 to 2.0), any subsequent calls made to that sub-process will fail, as the original sub-process with the initial version (1.0) will no longer be deployed, having been replaced by the new higher major numbered version.

See the following descriptions:

- how BPM handles versions at deployment. See [How BPM Handles Organization Model Versions at Deployment](#).
- when and how you should amend version numbers in your projects. See [Amending Version Numbers](#).

## Version Usage Best Practices

TIBCO recommends that you follow the guidelines in this topic for changing project versions.

- Change *qualifier* to identify different builds.
- Change *micro* to reflect and trace defect corrections, so that defect tracking can refer to the version correcting the defect.
- Change the *minor* version to reflect any process logic change that has no impact on the interface, so that a client application using the process as a service does not see any difference.
- Change the *major* version to reflect a change in the process interface.

See "WSDL Change Considerations for Application Upgrade" in *TIBCO Business Studio - BPM Implementation* for more information on what changes affect the process interface and what do not.

## How BPM Handles Organization Model Versions at Deployment

When an organization model is deployed to BPM, each major version is treated as a complete and separate organization model, and each minor version is treated as an additive update to the existing organization model with that major version number.



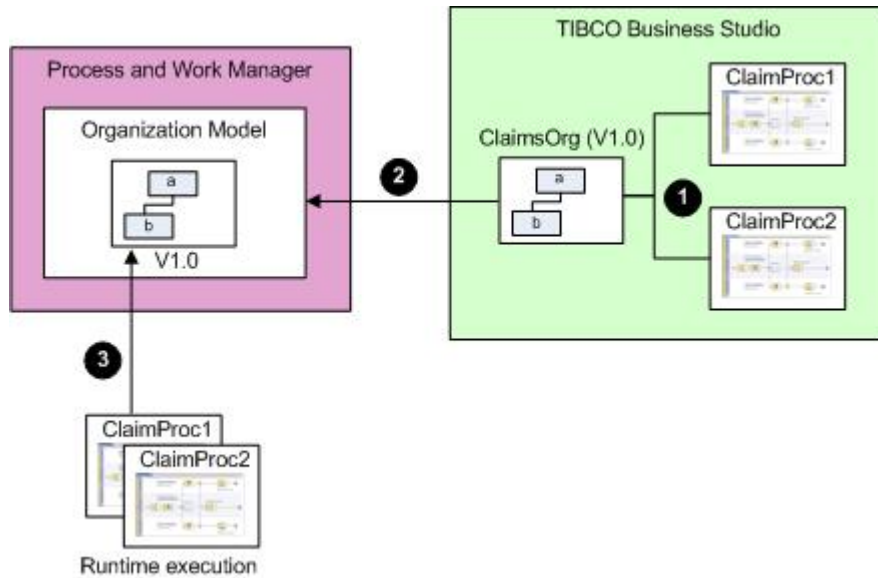
If an update to an organization model involves destructive or conflicting changes - for example, deleting an organization unit or position - you must give the organization model a new major version number.

At runtime, a process executes against the accumulation of all deployed organization models belonging to the **major** version referenced in its process definition. For example, an organization model may have been

deployed with version numbers, 1.1, 1.2 and 1.3. All processes that reference that organization will execute against organization model version 1.3.

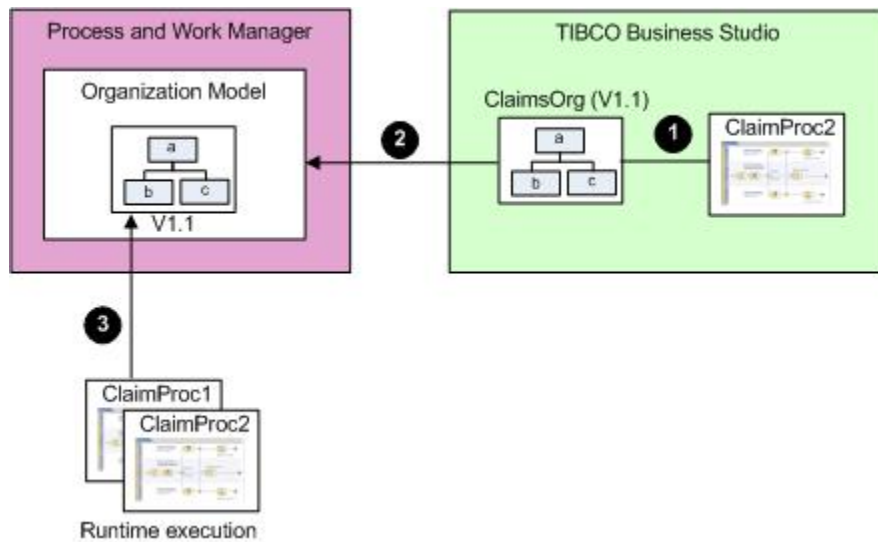
The following example shows how organization model versioning can be used.

EasyAs Insurance is rolling out a BPM implementation, starting with Customer Services.



1. The implementation defines two applications, ClaimProc1 and ClaimProc2, which involve departments a and b. These departments are modeled in the ClaimsOrg organization model, which is given a version number of V1.0. The major version number (1) is recorded in the ClaimProc1 and ClaimProc2 process definitions.
2. The ClaimsOrg organization model and both ClaimProc applications are deployed to BPM.
3. At runtime, both applications execute against V1.0 of the organization model.

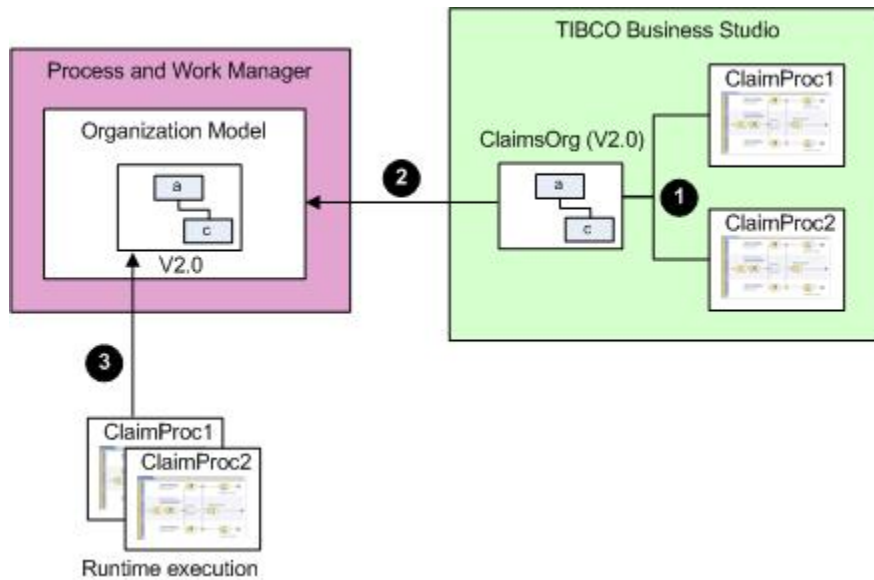
Following some user testing, EasyAs decide that they need to change one of the processes to involve an additional department.



The ClaimsOrg organization model is modified to include department c. As this is an extension to the existing organization model, the version number is incremented to V1.1.

1. ClaimProc2 is modified to use department c. ClaimProc1 does not involve department c, so does not need to be modified.

- The ClaimsOrg organization model and ClaimProc1 application are deployed to Process and Work Manager.
  - At runtime, both ClaimProc applications execute against V1.1 of the organization model.
- A company reorganization now occurs which results in department b being broken up.



The ClaimsOrg organization model is modified to remove department b. As this is a destructive change, the version number is incremented to a higher major version number, V2.0.

- ClaimProc2 is modified to remove its references to department b. ClaimProc1 must also be modified to reference the updated organization model (even though the application does not involve department b). The major version number (2) is recorded in the ClaimProc1 and ClaimProc2 process definitions.
- The ClaimsOrg organization model and both ClaimProc applications are deployed to Process and Work Manager.
- At runtime, both applications execute against V2.0 of the organization model.

### Amending Version Numbers

If you try to deploy to BPM a project that contains an organization model with the same version number as an organization model that already exists, then the deployment fails.



This does not apply to projects that contain other assets. For projects that contain other assets, the qualifier (which is a date/time stamp) means that each project that is deployed does have a unique version number. For example, you may deploy a project twice, the second time within seconds of the first, but each time the project would have a different date/time stamp and therefore a unique version number.

Therefore, if you want to make changes to an existing organization model and redeploy it, you must give the project that contains the organization model a new version number.



When you increase the major or minor version number (depending on whether the change is additive or destructive), the numbers you choose should be sequential. This is because a process executes against the accumulation of all deployed organization models belonging to the major version referenced in its process definition. For example, an organization model may have been deployed with version numbers, 1.1, 1.2 and 1.3. All processes that reference that organization will execute against organization model version 1.3.

You can do one of the following:

- Increase the major version number. Change the major version number if the changes are destructive.



Name changes are regarded as destructive, because the old names have been removed.

- Increase the minor version number. Change the minor version number if the changes are additive rather than destructive. However, process definitions that reference the organization model also reference the **major** version number of the organization model. This means that if you have changed the major version number of an organization model then, if you want your process definitions to execute against the new organization model, you must change the major version number of the projects that reference the new organization model so that their major version numbers match.

Note that:

- A process executes against the accumulation of all deployed organization models belonging to the major version referenced in its process definition. This means that if only the organization model's minor version has changed, you do not have to change the version numbers of the processes that execute against it.
- Processes that do not need to use the new organization model do not have to have their major version number changed but can continue to execute against the previous version of the organization model.

## Changing the Major Version Number of an Organization Model

### Procedure

1. Make the changes required to the organization model.
2. To check what project references you have for your project, do the following:
  - a) Right-click the project and select **Properties**. The Properties dialog is displayed.
  - b) Select **Project References**. The project references for the project are displayed.
3. Amend the major version number by right-clicking the project that contains the organization model and selecting **Properties > Project Lifecycle** .  
See "Project Lifecycle" in the *TIBCO Business Studio Concepts Guide* for more information.
4. You must then refactor the project so that all the assets within the project are using the new version number. To do this:
  - a) Right-click the project and select **Refactor > Project Lifecycle** . The **ProjectLifecycle** dialog is displayed.
  - b) In **Changes to be performed**, select the assets whose version number you want to change.
  - c) In **New Values**, make sure version number is correct and click **OK**. The changes are applied.
5. Redeploy the project. See [Deploying BPM Applications](#).
6. For any projects that reference that organization model, right-click and select **Properties > Project Lifecycle** . Amend the major version numbers to match the organization model's major version number
7. Redeploy the projects.



# Deploying BPM Applications

This topic describes how to deploy a BPM application for use at run-time.

Once you have developed a BPM application, its constituent elements must be deployed to the BPM runtime so that the application can be run. The following elements must be deployed:

- the process (or processes, including both business processes and pageflow processes),
- any organization model used by the process,
- any forms used by the process,
- any structured data used by the process.



You do not need to deploy the Business Object Model for a project separately, even if it is contained in a separate project from the one you are deploying. It will be included with the project using the Business Object Model by TIBCO Business Studio when it is deployed.

You can deploy an application to BPM in any of the following ways:

- From the ActiveMatrix Administrator:
  - Using the UI
  - Using the CLI

See [Deploying an Application from Administrator](#) for details.
- From TIBCO Business Studio. TIBCO Business Studio generates CLI scripts and runs them, so in some circumstances has the same effect as using Administrator CLI. The scripts report errors in case of failures. From TIBCO Business Studio you can:
  - Deploy an application to the server directly using a deployment server. You can use either a deployment wizard or a drag-and-drop method.
  - Export to a Distributed Application Archive (DAA) and deploy.
  - Create and execute a deployment script that can be used to deploy a project, or to deploy multiple projects simultaneously, from the command-line. See [Scripted Deployment](#) for information on this process.

## Deployment from TIBCO Business Studio Direct Or DAA Export

To deploy an application from TIBCO Business Studio to your BPM runtime, you have three options: Direct deployment, Export to a Distributed Application Archive (DAA) and deploy to a remote server, and create and execute a deployment script that can be used to deploy a project, or to deploy multiple projects simultaneously, using the command-line.

- Deploy directly to the BPM runtime using a deployment server. To do this:
  1. Create and connect to a deployment server. See [Creating a New Deployment Server](#).
  2. Deploy to the server using either a deployment wizard or a drag-and-drop method. See [Deploying an Application Using the Deployment Wizard](#) and [Deploying an Application by Dragging and Dropping](#) for details.
- Export to a Distributed Application Archive (DAA) and deploy to a remote server. See [Exporting to DAA](#).
- Create and execute a deployment script that can be used to deploy a project, or to deploy multiple projects simultaneously, using the command-line. See [Deploying BPM Applications](#) for information on this process.

## Direct Deployment

Direct deployment from TIBCO Business Studio has the following goals and characteristics:

- It is intended to support the solution designer in rapid, iterative development in either a development or a test environment.
- TIBCO Business Studio generates a DAA and deploys it to the BPM runtime. However, the generated DAA is not available in the **Exports** folder of the project.

## Exporting a DAA

Exporting a DAA file to ActiveMatrix Administrator for deployment is intended for production use, where finer-grained control over whether deployment is forceful or not, is required. The DAA is typically available in the **Exports** folder of the project.



Note that DAAs generated using the scripted deployment facilities (see [Deploying BPM Applications](#)) are not found in the **Exports** folder of the originating project, but are generated as part of the deployment project.

## Deploying a Process That Exposes a Web Service

When you deploy a project that exposes a web service you must bind the system participant to the appropriate HTTP Connector resource instance in the BPM runtime.

- The system participant defines the web service endpoint.
- The HTTP Connector resource instance is used by BPM to provide external client applications with a runtime connection to the web service.

You can perform this binding using the **Property Configuration** page of the **DAA Deployment Wizard** wizard.

For more information see [Deploying an Application from TIBCO Business Studio](#) and *TIBCO Business Studio - BPM Implementation*.

Alternatively, you can export the project to a Distributed Application Archive (DAA), then use the Administrator interface in the BPM runtime to perform the binding and deploy the application. See the Administrator interface documentation for your BPM runtime environment for more information.

A client application (which can be another process) hosted in the BPM runtime can now call the exposed web service operation on its virtualization binding or, if necessary, on its SOAP binding.

An external client application will need to access the exposed web service on its SOAP binding, using a concrete WSDL.

You can generate a concrete WSDL for the application from the Administrator interface in the BPM runtime.

## Deploying a Global Signal Definition Project

You must deploy a Global Signal Definition project before trying to deploy any process project/s that throws a referenced global signal, or a process that catches a global signal.

### Procedure

1. Deploy the Global Signal Definition project in the same way you would deploy a Business Studio project.
2. Deploy the project/s that throw the referenced global signal or catch one.  
You will then be able to reference the global signals in multiple projects.



See [Re-Deploy and Upgrade Global Signal Definition Run-time Application](#) for issues you should be aware of before you deploy your project.

## Business Process Execution Language (BPEL) and Deployment

Processes are deployed and executed using an enhanced version of BPEL 2.0. Conventional BPEL (WS-BPEL) is an execution language that defines how services can be orchestrated in order to provide further, higher level, services. Although it is a very powerful method for defining certain classes of Straight-Through Processing (STP) process, it is not sufficiently flexible to deal with all the requirements of a business process management process. For example, the language contains no definition of a user task, and cannot represent all required workflow patterns.

TIBCO ActiveMatrix BPM therefore implements a superset of BPEL 2.0, which contains enhancements and extensions to support the required business process management functionality and workflow patterns.

For more information about WS-BPEL 2.0 see the OASIS website at <http://www.oasis-open.org/home/index.php>.

## Preparation for Deployment

Before you deploy a project, you need to ensure that you have the correct TIBCO Business Studio capabilities switched on, and that all elements of the project are ready to be deployed.

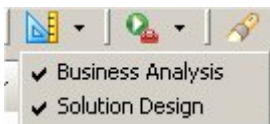
### Solution Design Capability

You must have the Solution Design capability turned on to be able to create and use deployment servers. If it is not turned on, enable the Solution Design capability from the toolbar.

Click the dropdown next to the **Enable/Disable Business Studio Capabilities** icon.



Select **Solution Design**.



### Check Project Elements

Check the following elements of your project:

- **Validation errors:** Ensure that no validation errors are displayed that relate to the project you wish to deploy. It is not possible to deploy a project, or to create a Distributed Application Archive (DAA) for export, if there are any validation errors.
- **Scripts:** Any scripts in a process must be implemented as JavaScript before the process is deployed to the BPM runtime.

See "Implementing Script Tasks" in *TIBCO Business Studio - BPM Implementation* for instructions on how to do this.

- **Forms:** If you require any of your user tasks to use forms other than the default forms, these must be created before you deploy. See the TIBCO Business Studio Forms documentation for further details.



You do not have to create any forms for user tasks before you deploy a process. (You can obviously do so if you wish, but it is not required.) At runtime, a default form will be used for any user task that does not have a form explicitly created for it.

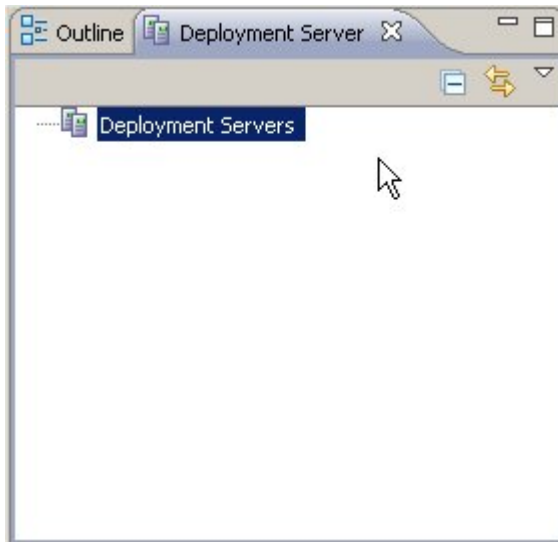
- **Versions:** Ensure that the version numbers of the projects and artifacts that you are deploying are set correctly. This is especially important for projects containing organization models. See [About Project Versions](#) for details.

## Creating a New Deployment Server

Before you can create a deployment server, you need to know the URL on which the server will run. If necessary, consult your system administrator to obtain this information. The BPM application is called `amx.bpm.app` by default, but because there may be multiple instances of it, the name can vary.

### Procedure

1. Click the **Deployment Server** view.



This view is by default displayed at the bottom left corner of the TIBCO Business Studio window. If it is not visible, select **Window > Show View > Other...** and in the Show View dialog, expand **Studio**.

2. Click **Deployment Server** and then **OK**.
3. Right-click **Deployment Servers**, then select **New > Server**.
4. On the Server Name and Runtime dialog:
  - a) Enter a suitable server name.
  - b) From the **Runtime** dropdown, select the type of runtime server to which you want to connect and deploy applications:

**Administrator Server** to deploy your application.

**WebDAV Server** to deploy your project documentation.

5. Click **Next**.

The Runtime Server Parameters dialog is displayed. The parameters on the Runtime Server Parameters dialog depend on the runtime server selected.

When you select the runtime **Administrator Server**, the Runtime Server Parameters dialog displayed is as shown.

**New Server**

**Runtime Server Parameters**  
Set parameters of the runtime server.

Runtime Server Parameters:

Server URL:

Username:

Password:

Use form-based authentication:

Hide System Applications:

Default Environment:

Show Only Default Environment:

Default Application Folder:

Default Target Application:

Save password

Saved passwords are stored on your computer in a file that is difficult, but not impossible, for an intruder to read.


Repository:


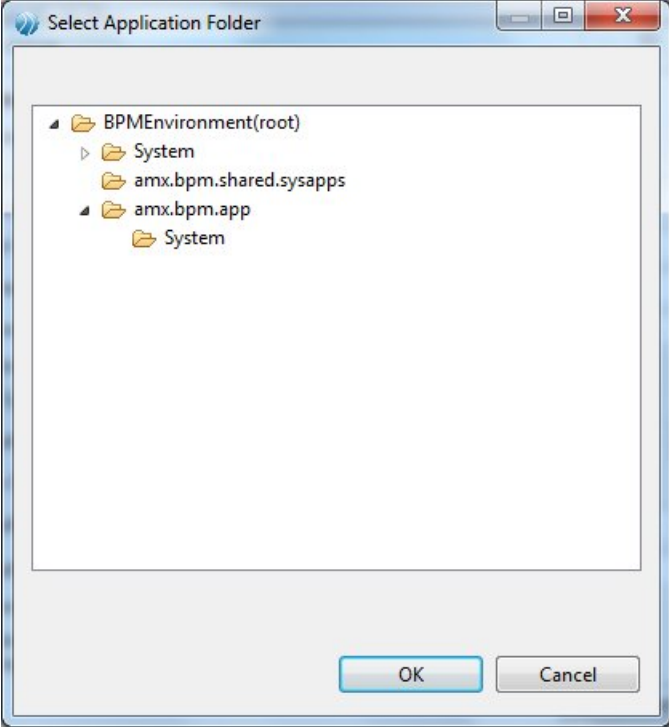
Repository Type:

When you select the runtime **WebDAV Server**, the Runtime Server Parameters dialog displayed is as shown.

The following table describes the parameters in the Runtime Server Parameters dialog:

Field	Description
Server URL	<p>The login URL for the Administrator interface in the BPM runtime. This URL is:</p> <p><i>protocol://host:port</i></p> <p>where:</p> <ul style="list-style-type: none"> <li><i>protocol</i> is the communications protocol being used by the Administrator interface in the BPM runtime, either <b>http</b> or <b>https</b>. This is determined during the BPM runtime installation.</li> <li><i>host</i> is the DNS name or IP address of the server hosting the BPM runtime.</li> <li><i>port</i> is the port being used by the Administrator interface in the BPM runtime. The default value is <b>8120</b>.</li> </ul>
Site URL	<p>URL for the WebDAV server. Typically, this URL is</p> <p><i>protocol://host</i>, where:</p> <ul style="list-style-type: none"> <li><i>protocol</i> is the communications protocol being used by the WebDAV server.</li> <li><i>host</i> is the DNS name or IP address of the server hosting the WebDAV server.</li> </ul> <p>If you do not know your WebDAV server URL, contact your system Administrator.</p>
Username	The username needed to log in to the Administrator interface in the BPM runtime.
Password	The password associated with the <b>Username</b> specified.

Field	Description
Use form-based authentication	<p>Check this box to support form-based authentication. By default, the checkbox is not selected.</p> <p>If this is selected, the Administrator server validates your username and password and then creates a session identified by a unique key. This key is passed between the client and server on each subsequent HTTP request. See the TIBCO ActiveMatrix Administrator documentation for further details.</p>
Hide System Applications	<p>Checkbox to hide (or show) the system applications in the list of applications in the selected environment. By default, the checkbox is selected.</p> <p>You can choose to view the system application by unchecking the checkbox.</p>
Default Environment	<p>The default environment on which the BPM applications are to be deployed. By default, the following environments are listed:</p> <ul style="list-style-type: none"> <li>• BPMEnvironment</li> <li>• DevEnvironment</li> <li>• SystemEnvironment</li> </ul> <p>Of these, <b>DevEnvironment</b> is by default selected. Select the environment where your copy of the BPM application is located. By default, this is <b>BPMEnvironment</b>.</p> <p>Click the <b>Refresh</b> icon  to list the environments available on the selected server. If for example you have multiple instances of the BPM application each in its own environment, all those environments will be listed for you to select from.</p>
Show only Default Environment	<p>Checkbox to display in the <b>Administrator Explorer</b> view all the available applications and nodes on only your specified <b>Default Environment</b>. This option is checked by default.</p> <p>The <b>Administrator Explorer</b> view allows you to browse the Administrator servers when they are connected. To open the Administrator Explorer view, right-click the deployment server created and select <b>Show Administrator Explorer View</b>. See <a href="#">Accessing Runtime Clients and Administrator Server from TIBCO Business Studio</a> for additional information.</p> <p>If you deselect the checkbox, all the environments available on the BPM runtime, along with the applications and nodes on each of them, will be listed.</p>

Field	Description
Default Application Folder	<p>The name of the folder in which the application will be deployed. Initially this field is empty.</p> <p>Click . The Select Application Folder dialog is displayed, listing the available folders in the <b>Default Environment</b> you have selected.</p>  <p>Select the desired folder and click <b>OK</b>.</p> <p>See "Application Folders" in <i>TIBCO ActiveMatrix BPM SOA Administration Software</i> for more information about application folders.</p>
Default Target Application	<p>The name of the default target application to which applications will be deployed. By default, the value is set to the default name of the BPM product application, <b>amx.bpm.app</b>.</p> <p>If your installation includes more than one ActiveMatrix BPM system, that is more than one instance of the BPM application, you must specify here the instance to which you want to deploy your projects. You will need a separate deployment server for each separate instance of the BPM application. See ...</p>
Save password	<p>Saves the password in a file on your computer. If this is not checked, you will get an error when you try to connect to the server.</p>
Repository Type	<p>The repository is where applications are staged before they are deployed. Accept the default <b>Workspace</b> for the Administrator interface in the BPM runtime.</p>

6. Click **Test Connection**.

A "Test Connection Successful" message is displayed, indicating that TIBCO Business Studio can successfully communicate with the server.





If the connection is not successful, the system keeps retrying and no message is displayed. If it continues to be unsuccessful, an error message is eventually displayed.

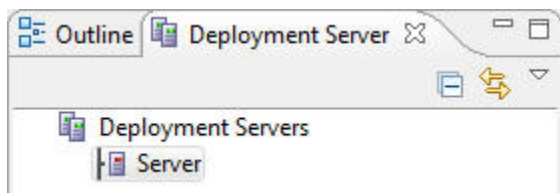
#### 7. Click **Finish**.



If the connection is to a BPM runtime that is configured to use secured (https) communications, the connection will fail unless you have already registered the self-signed certificate provided by the BPM runtime with the TIBCO Business Studio JRE. See [Enabling Deployment Over Secured \(https\) Communications for a Deployment Server](#).

### Result

The new deployment server now appears in the Deployment Server view. The deployment server is in a Disconnected state when created. You must connect to the server before you can deploy an application, or view the environments, applications, and nodes available on the server.



You can change the parameters for the server from the Deployment Server view. See [Viewing and Changing Deployment Server Properties](#). You can either restore the default settings for a server, or edit those you have already created.

### Local Development Server

When you create a new workspace in TIBCO Business Studio BPM Edition, a default deployment server configuration **Local Development Server** is created. This server is automatically configured correctly for a local development server with a default configuration.

The default Local Development Server is similar to any user-created deployment server. Its configuration can be modified and it can be deleted if not required.

### Enabling Deployment Over Secured (https) Communications for a Deployment Server

If you create a deployment server for a BPM runtime that is configured to use secured (https) communications, you must obtain the self-signed certificate provided by the instance hosting the BPM runtime, then register that certificate with the TIBCO Business Studio JRE.

Until you do this, you cannot deploy applications to this BPM runtime. This is because the TIBCO Business Studio JRE needs the certificate to be able to establish a secured connection to the deployment server.

#### Procedure

1. Contact your BPM runtime administrator to obtain the required certificate.
2. Copy the certificate from the BPM runtime node to a file (for example, `c:\instance1_cert.txt`) on the computer where you are running TIBCO Business Studio.
3. Make sure that you have the `JAVA_HOME\bin` directory in your command path.
4. Run the following Java `keytool` command to register the instance's certificate with the TIBCO Business Studio JRE:

```
keytool -import -trustcacerts -alias alias
        -keystore TIBCO_HOME\tibcojre\1.8.0\lib\security\cacerts
        -storepass passwd
        -noprompt -file cert_file
```

where:

- *alias* is a unique name (case-insensitive) to identify this BPM runtime.
- `TIBCO_HOME` is the folder where you installed TIBCO Business Studio.
- `passwd` is the password associated with this keystore. (The default value is "changeit".)
- *cert\_file* is the fully-qualified name of the certificate file you created in step 1 above.

The following response should be displayed:

Certificate was added to keystore

5. Restart TIBCO Business Studio (if it was running).

### Result

You can now use the deployment server to deploy applications to this BPM runtime.

## Connecting to a Deployment Server

You must connect to the deployment server before you can deploy an application.

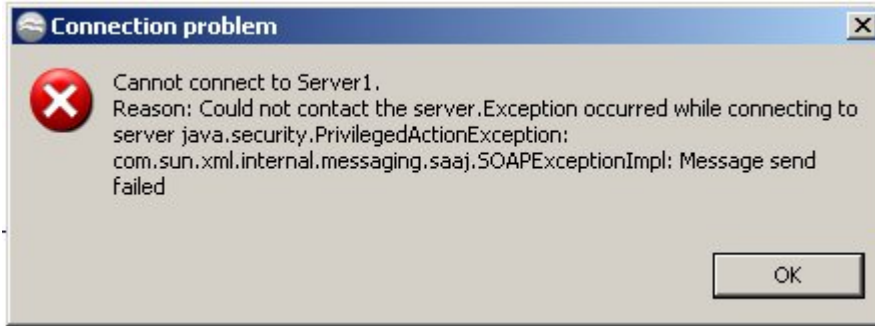
### Procedure

1. Go to the Deployment Server view, and double-click **Deployment Servers** to show all available deployment servers.
2. Right-click on the deployment server you wish to connect to, and select **Connect**.
3. If you are able to connect to the server, the Refreshing Administrator Server dialog displays briefly and then the Server State changes to **Connected** in the Properties view.

### Result

Deployment Server		
Advanced	Property	Value
	▲ Base	
	Name	Server
	▲ Server	
	Client Runtime	
	Runtime	Administrator Server
	Server State	Connected

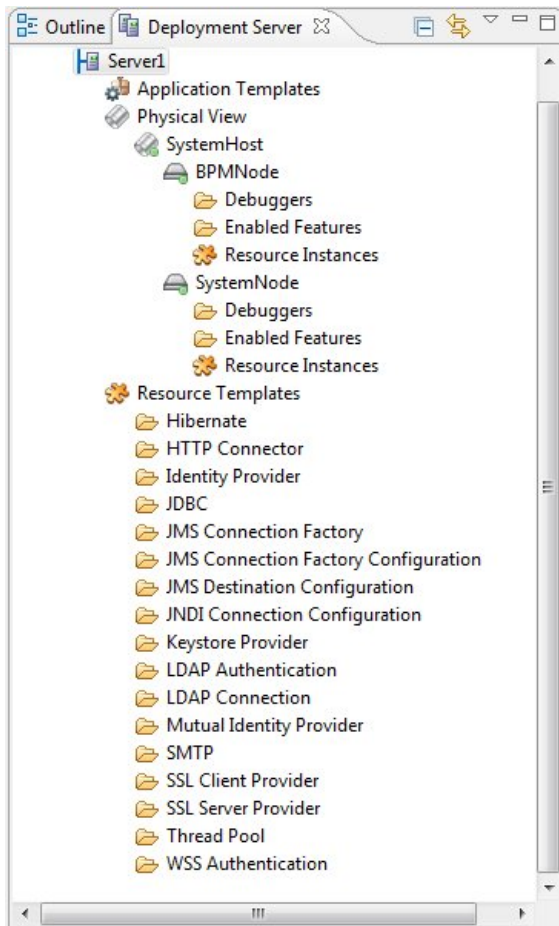
If you cannot connect to the server, you will see an error message.



You could try the following steps to investigate the cause of the failure to connect:

- Make sure that the server to which you are trying to connect is switched on.
- Check that the parameters of the deployment server, particularly the IP address and port to which it is trying to connect, are correct.

The following figure displays the Deployment Server view for a server created with default values.



If you unchecked the Show Only Default Environment checkbox while creating the deployment server, the BPMEnvironment (or whatever default you selected) and any other environments available on the server are displayed.

Expand the BPMEnvironment to see all the applications and nodes that are already deployed to the BPM Environment on the server. Expand the Applications to list the applications deployed to the server,

including the target BPM application (by default **amx.bpm.app**). Expand the Nodes to list the nodes deployed to the server, including the BPMNode.

### Refreshing a Deployment Server

Occasionally, you may need to manually refresh the deployment server in TIBCO Business Studio to synchronize with the Administrator server. Typically, this is necessary after resource instances and/or non-BPM applications (for example, Mediation) with virtual bindings are deployed from *outside* TIBCO Business Studio. If you deploy a BPM application with a dependency on any components that were deployed outside TIBCO Business Studio, the binding/wiring dialog does not list those components.

### Disconnecting From a Deployment Server

#### Procedure

1. Go to the Deployment Server view, and double-click **Deployment Servers** to show all available servers.
2. Right-click on the server you wish to disconnect from, and select **Disconnect**.
3. No message is displayed. The Server State changes to **Disconnected** in the Properties view.

### Viewing and Changing Deployment Server Properties

#### Procedure

1. Go to the Deployment Server view, and double-click **Deployment Servers** to show all available servers.
2. Right-click on the server you are interested in, and select **Properties**.
3. You can edit the **Server General** settings; or click the **Restore Defaults** button if you want to use the system default settings.
4. Click **Test Connection** from the **Server General** settings. A **Test Connection Successful** message is displayed, indicating that TIBCO Business Studio can successfully communicate with the server.
5. Click **OK** to accept the settings.

### Accessing Runtime Clients and Administrator Server from TIBCO Business Studio

You can access the BPM runtime clients - Openspace and Workspace - from TIBCO Business Studio.

#### Procedure

1. Go to the **Deployment Server** view. Ensure that at least one server has been created.
2. Right-click on the server you are interested in and select **Open**. The following options are listed:
  - Open Administrator in web browser
  - TIBCO Openspace
  - TIBCO Workspace
3. Select the client from the options to open it in TIBCO Business Studio.



You cannot configure the URL of the runtime client you wish to open from within TIBCO Business Studio. The system assumes that the URL (protocol and hostname) of the runtime clients is the same as that of the Administrator server.

The system cannot open the clients even if the protocol used by the Administrator server and the clients differs. For example, if the URL for the Administrator server is at `https://localhost:8120/` and the URL for Openspace is `http://localhost:8080/openspace`, Openspace cannot be opened within TIBCO Business Studio.

## Deploying an Application from TIBCO Business Studio

You can deploy an application from TIBCO Business Studio to the BPM runtime either by using a deployment server or by exporting to a Distributed Application Archive (DAA).

Do not deploy multiple application instances from the same application template.



If a project references artifacts contained in another project - such as an organization model - the referenced project must be deployed beforehand for the application to run successfully.

However, if your project references a BOM which is contained in another project, you need not deploy the project containing the BOM beforehand.

Deployment and undeployment of an application can be done from the Administrator UI, the Administrator CLI, or from TIBCO Business Studio. TIBCO Business Studio generates CLI scripts and runs them, so in some circumstances has the same effect as using Administrator CLI. The scripts report errors in case of failures.

## Deploying an Application Using the Deployment Wizard

You can deploy an application to a BPM node by deploying its project (or the Distributed Application Archive (DAA)) to the appropriate deployment server.

If a project references artifacts in other projects, the referenced projects must be deployed before proceeding to deploy the project.

If the project has bindings or wiring defined, you can configure them through the wizard as part of the deployment procedure.

### Procedure

1. In the Deployment Server view, select the deployment server you require.




You may have only one deployment server. If you have more than one instance of the BPM application, you need a separate deployment server for each instance. See [Creating a New Deployment Server](#).

2. Connect the deployment server to the BPM server (if you have not already done so).
3. Right-click the deployment server you want, and select **Deploy Module...**
4. Choose either **Deploy BPM Project** or **Deploy DAA...** from the Select Module Type dialog, then click **Next**.

**Deploy DAA...** enables you to deploy a previously exported DAA. For example, someone may have exported their project to a DAA (described in [Exporting to DAA](#)) and emailed the DAA to you.

- If you selected the **Deploy DAA...** option, the Application Configuration dialog is displayed. Click **Workspace...** to select the DAA from the workspace; or click **File System...** to select the DAA from the local file system. Browse and select the DAA to be deployed and click **Next**. Skip [Step 5](#) and go to [Step 6](#).
- If you selected the **Deploy BPM Project** option, the Select BPM Project dialog is displayed. All the supported projects available in the workspace are listed.

Select the project to be deployed and click **Next**.

5. On the Application Configuration dialog, either accept the default values in the **Environment Name** and **Application Folder** fields (these fields are populated with values specified when you set up the deployment server; see [Creating a New Deployment Server](#)), or if you want to deploy to another destination, perform these substeps:
  - a) In the **Environment Name** field, select the required environment from the drop-down menu displayed.
  - b) In the **Application Folder** field, click  and select your required folder from the list.
  - c) Click **Next**.



The checkbox **Upgrade Existing Application** is disabled if the application is being deployed for the first time. For subsequent deployments, the checkbox is enabled and selected by default. See [Deploying the Organization Model](#) for details on configuring the behavior when upgrading existing applications.

**Refresh Server Timeout** specifies the duration in seconds for which TIBCO Business Studio waits for responses from the BPM runtime.

The **Resolve Mode** checkbox applies to TIBCO SOA projects. Keep the checkbox clear (unchecked) for BPM application deployment. This field specifies if dependencies on target product applications have to be checked. When selected, the dependencies are checked and nodes where the applications are deployed are restarted to load software updates. This is required when downgrading a feature or applying extension implementations for TIBCO SOA projects.

If the project or DAA contains properties that can be configured, the Property Configuration dialog is displayed with the application properties in a tabular view.

**Property Configuration**  
Configure application properties.

Name	Type	Value
DefaultEmailSender1	SmtpConfiguration	WPEmailChannelSMTPSharedResource -> %%DefaultEmailSender1%% (Existing Resource Instance)
HttpInboundConnectionConfig	string	%%BPMSample_BPMSampleProcess%% -> httpConnector

Buttons: < Back, Next >, Finish, Cancel, Override...

6. Select a property and click **Override...** to change the configuration.



During deployment, do not change the BDS Application deployment BDSCaseDataStoreResource configuration property value.

By default, the properties are mapped to substitution variables, which provide a level of indirection. To override the mapping:

- a) Double-click the Property Value field, or select the property and click **Override...**. The dialog to override the selected property appears.

**Override value of 'HttpInboundConnectionConfig' property**

Select mapping to existing resource instance or define how the resource instance should be created:

Existing Resource Instance:

Name: httpConnector

Map through Substitution Variable

Substitution Variable: %% BPMSample\_BPMSampleProcess %%

Buttons: ? (help), OK, Cancel

- b) To map to an existing resource instance, choose the name of the existing resource instance from the drop-down list.
- c) The **Map through Substitution Variable** checkbox is selected by default. Uncheck if you do not wish to use substitution variables.



See *TIBCO ActiveMatrix Composite Development* guide for additional information about substitution variables.

If the project or the DAA contains unresolved promoted references, the Wiring Configuration dialog is displayed.

To configure the wiring:

- a) Click **-add wire-** in the row containing the unresolved promoted reference. A drop-down listing the available service virtualizations and endpoints appears.



- b) Select the virtualization or endpoint that you want to wire the promoted reference with and click outside the text.

Repeat these substeps for all the unresolved promoted references listed in the dialog.

7. Click **Finish**.

A progress dialog displays the execution status while deployment takes place. The status messages indicate whether the deployment has finished with no errors, or if any errors occurred. The progress messages can also be seen on the **Console** tab.

When the deployment finishes with no errors, the application appears in the **Deployment Servers** view.

8. Expand the **Deployment Servers** view and select the Applications node. A refresh may be needed for the deployed application to appear in the list (right click on the server name and select **Refresh** or just hit **F5**).



See [Deployed Applications](#) for information on how the application name is created.

## Deployment of Global Signal Definition Project and Process with Global Signal Events

It is possible to deploy a process with global signal events without deploying the Global Signal Definition project. However, if you do this, then the process will halt with an error at runtime.

You should deploy the Global Signal Definition Project before you deploy processes which contain global signal events.

## Deploying an Application by Dragging and Dropping

You can also deploy a project (or a previously exported DAA) by dragging it to a deployment server and dropping it.

### Procedure

1. Select a project (or a previously exported DAA) in the Project Explorer.
2. Drag the project (or a previously exported DAA) to your deployment server in the Deployment Server view, and drop.

When you drop a project, the Select Project dialog displays with the project name highlighted. Proceed from [Step 4](#) described in [Deploying an Application Using the Deployment Wizard](#).

When you drop a previously exported DAA, the Application Configuration dialog displays. Proceed from [Step 4](#) described in [Deploying an Application Using the Deployment Wizard](#).

## Verifying the Deployment

### Procedure

1. Open the Administrator interface in the BPM runtime to check that the application has deployed correctly. You can do this in the Deployment Servers view by right-clicking on the deployment server you are using and selecting one of the following:
  - **Open > Open Administrator in web browser**. Log in to view the Administrator interface in your BPM runtime. Click **Applications** to display the list of available applications.
  - **Show Administrator Explorer View**. In the Administrator Explorer, expand the Applications node under the environment on which you deployed the application.
2. Check that your application has deployed successfully. If you view the Administrator UI from a web browser, the **Runtime State** should be **Running** and the **Action History** should be **Deploy with Start Successful**.



## Result

See the Administrator interface documentation for your BPM runtime environment for details of how to use the Administrator user interface.

## Exporting to DAA

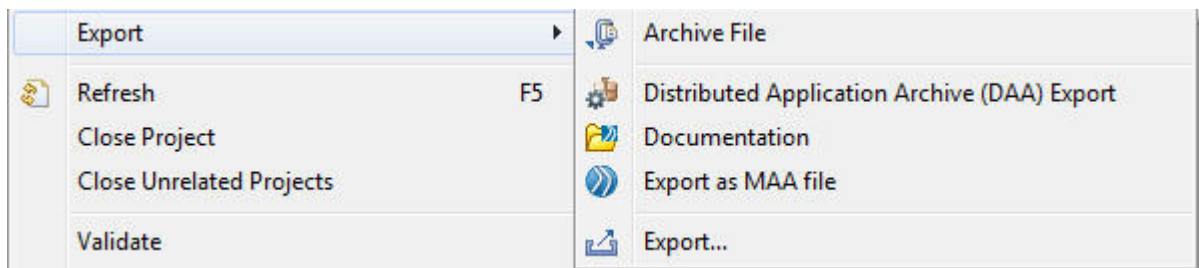
You can export a project to a Distributed Application Archive. This is a suitable method for situations where you do not have authorization to export directly to the BPM runtime; for example, in a production environment where developers cannot deploy directly to a production server but can deliver the project as a DAA archive to be uploaded by those responsible for the production machines.

See the Administrator interface documentation for your BPM runtime environment for information on uploading DAA archives and for the completion of the deployment process using the Administrator interface in the BPM runtime.

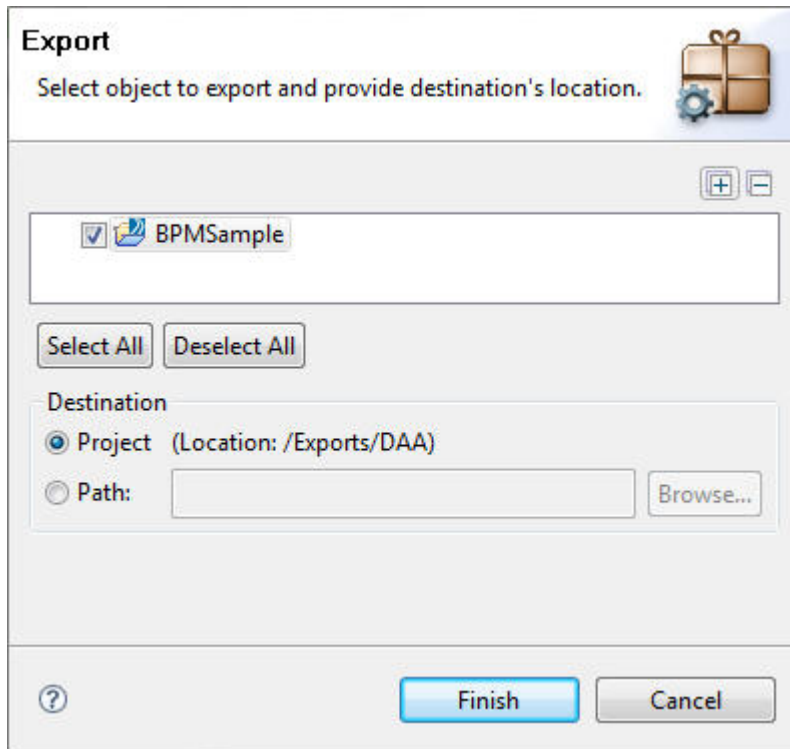
Exporting to DAA may also be useful for troubleshooting.

## Procedure

1. Right-click the project in Project Explorer and select **Export > Distributed Application Archive (DAA) Export**.



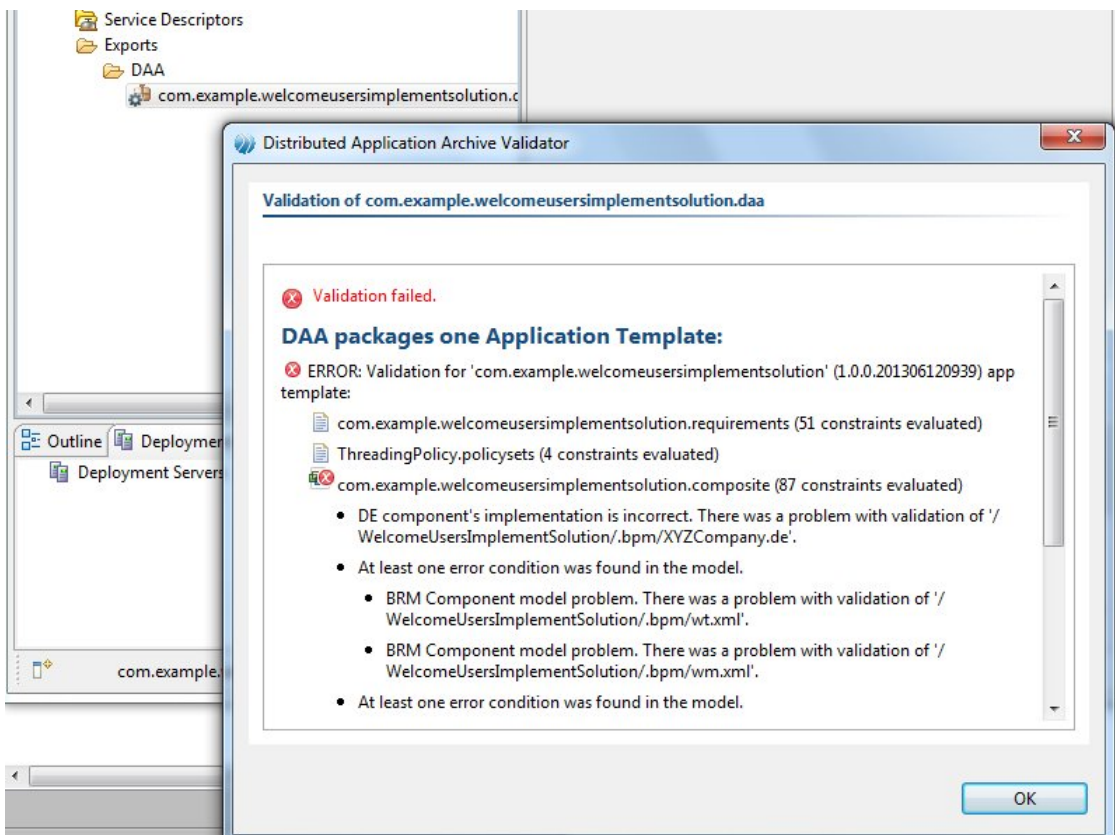
2. The export wizard is displayed. Make sure that the correct project is selected. In the **Destination** pane the default choice **Project** specifies that the DAA should be sent to an **Exports** folder within the project. To export the DAA to a different location, select **Path** and specify the folder location.



3. Click **Finish**. A progress meter is briefly displayed in the status bar. The DAA file is visible in the location specified.

## DAA Validation

You can see the contents of the DAA file by right-clicking on the file and selecting **Validate DAA**.



TIBCO Business Studio runs a series of checks on the generated DAA files. The DAA could be considered valid by TIBCO Business Studio, but not all requirements can be catered for. For example, a BPM project could be dependent on an Organization model. If that Organization Model is not deployed, the deployment of the BPM project will fail even though the DAA is considered valid.

Another example of this is project where the bindings need to be configured in the runtime environment. A "Deploy with Start" on such a project may fail, but it may be possible to successfully deploy the project, configure the bindings, and then start the application.

## Deploying an Application from Administrator

See "Deploying Applications" in *TIBCO ActiveMatrix BPM - SOA Administration* for information on deploying via TIBCO ActiveMatrix Administrator, using either the GUI or the CLI interfaces.

### Files Generated by Business Studio

When you deploy a BPM or an SOA project from TIBCO ActiveMatrix Administrator, the deployment uses scripts generated by TIBCO Business Studio.

TIBCO Business Studio generate the following three files as the basis of the deployment process:

- `<deployment-name>.deployment-build.xml`
- `<deployment-name>.deployment-config.xml`
- `<server-name>.properties`

#### **`<deployment-name>.deployment-build.xml`**

An ant script file that controls the CLI process. This generally contains some of the following ant targets:

- `undeploy.app` — Undeploys the application
- `upgrade.app` — Upgrades the application
- `delete.app` — Deletes the application
- `upload.daa` — Uploads a new instance of the DAA
- `edit.properties` — Modified any properties or configuration of resource instances
- `wire.application` — Wires any connections that the application will use
- `distribute.app` — Distributes the application
- `deploy.app` — Deploys the application to the specified target nodes
- `start.app` — Starts the application

In practice this files is limited in its reuse and is not used by the automated deployment process.

#### **`<deployment-name>.deployment-config.xml`**

This configuration file describes the application, components, end-points and references that will be used for a given application. The configuration file is the key to the deployment process. It is vital that this file correctly describes the configuration of the application to be deployed. The file must be regenerated whenever the external configuration of an application changes. Typically this includes:

- Adding or removing promoted references to an SOA project
- Adding or removing promoted services to an SOA project
- Adding or removing end-points to BPM processes. These are typically new participants of type "System".

### <server-name>.properties

A properties file containing the URL and admin name and password of the server to which the DAA should be deployed. Strictly this does not need to be provided to the configuration team as the file is easy to regenerate.

This contains the following entries:

- adminURL — The URL of the AMX administrator on which the deployment should occur
- username — The name of an AMX administrator account
- password — The password of the AMX administrator account
- httpConnectionTimeout — Connection timeout in milliseconds for administrative operations

## Deployed Applications

This topic covers application components and application names after deployment.

### Application Components

When a project containing at least a process or an organization model is deployed, it creates:

- An application template.
- An application instance of the template

The application instance is "wired" to other resources on the ActiveMatrix BPM server. This defines the relationship between the developed application and the other applications and resources on which it depends.

### Application Names

When an application is deployed, it must be identified by a unique name. In a development or testing environment, there may be several versions of an application deployed together, so it is necessary to distinguish between them.

When deploying an application, you can choose an application name either from the Application Configuration dialog in the **DAA Deployment Wizard** in TIBCO Business Studio, or from the TIBCO Administrator UI.



During deployment, do not change the BDS Application deployment `BDSCaseDataStoreResource` configuration property value.

A typical BPM application name is constructed from the project ID and is of the form **com.example.projectname**, where *projectname* is the project name with any internal spaces removed. You can replace **com.example** by your own organization's domain name, or whatever else is required by your naming conventions.



The project ID is assigned when you create the project, and you can change it subsequently by right-clicking the project and selecting **Properties > Project Lifecycle**.

You can change the default domain name by selecting **Window > Preferences > User Profile**. Then edit the **Domain Name** field which is set as `com.example`.

It is important to realize that the deployed application name may not necessarily bear any direct resemblance to the project name in TIBCO Business Studio. For an example about deploying and undeploying applications, see [How BPM Handles Organization Model Versions at Deployment](#)

## Deploying the Organization Model

To be used at runtime an organization model must be deployed to a BPM server. In BPM an organization model is seen as part of an application. The application consists of a business process and any supporting

material, which can include an organization model. One organization model can be used by multiple applications; your business might have different applications for different business functions, but all of them would need to reference a model of the same organization.

When you deploy an organization model, any Resources that you have defined are not deployed (with the exception of the Human Resource Type, which must exist and is always deployed). All other parts of the organization model as defined in TIBCO Business Studio are deployed. For further details, see "Resources (Users)" in the *TIBCO ActiveMatrix BPM Concepts* guide.

### The Organization Model at Runtime

At runtime in BPM, how an end user's position is defined in the organization model can be used to determine what type of work is presented to them. Customized role-based clients can offer work to users depending on the Position they hold, the Capabilities or Privileges attributed to them, or both. For example, a user with an 'LDAP Administration' privilege could be offered all and only LDAP work.

## Deploying a BPM Application Fails If the Referenced Organization Model Has Been Upgraded

Business Process applications fail to deploy and go into 'Waiting for dependencies' state when the referenced Organization Model version has been upgraded.

TIBCO ActiveMatrix Administrator identifies applications by their name and version number. When an Organization Model is deployed with the same name as that of an existing Organization Model, ActiveMatrix Administrator undeploys the existing one, regardless of the version. As Business Processes reference the major version of the Organization Model, removing the referenced Organization Model results in a failure to deploy the Business Process application.

### Possible Solutions

- Upgrade the Business Process applications to refer to the newer Organization Model version and redeploy the affected applications.
- Deploy the upgraded Organization Model with a new Application name. This ensures that ActiveMatrix Administrator does not undeploy the older organization model. At runtime, one can reference entities from both organization models.

## Deployment to Multiple BPM Systems

From version 2.2, ActiveMatrix BPM enables you to install multiple BPM systems - that is, multiple instances of the BPM application - in the same ActiveMatrix enterprise.

Multiple BPM instances may be installed:

- in the same ActiveMatrix environment, or in separate ActiveMatrix environments.
- on the same TIBCO host.
- on the same physical machine.

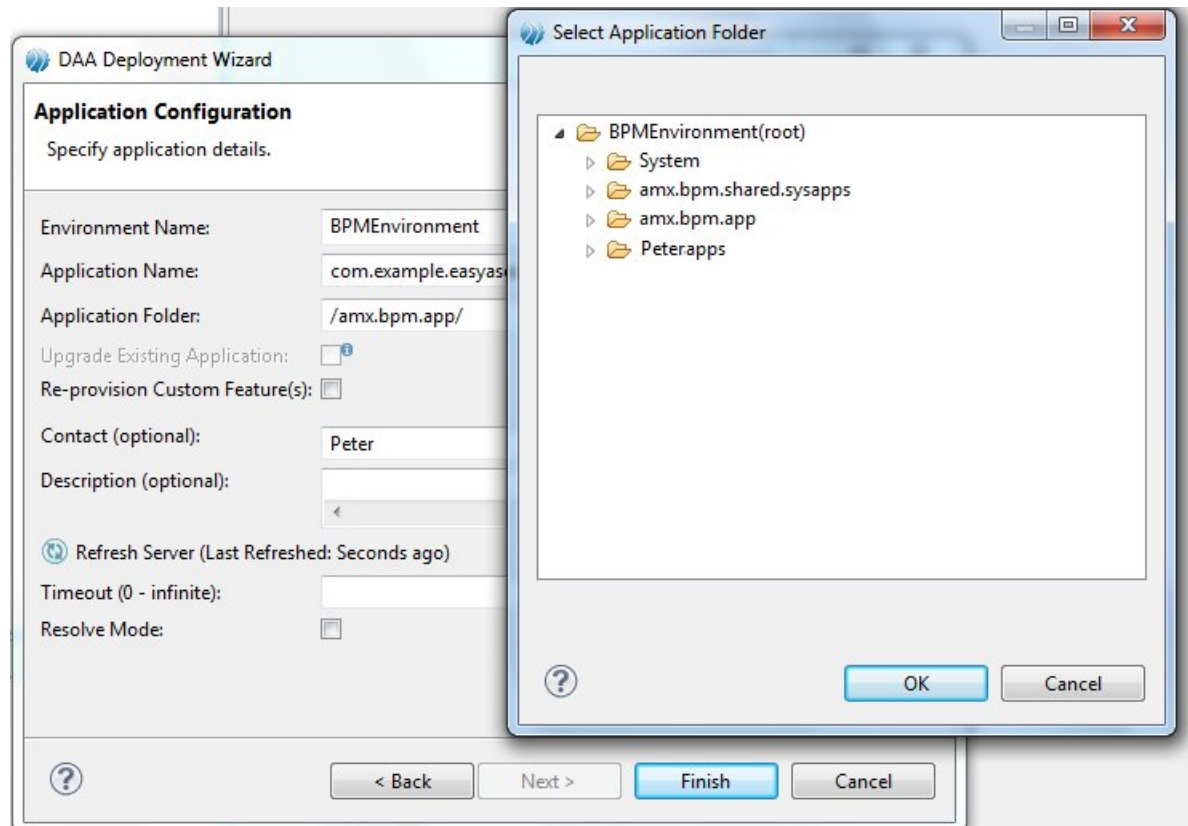
See "Multiple Installations of BPM Systems in the Same ActiveMatrix Enterprise" in *TIBCO ActiveMatrix BPM Installation and Configuration*.

If you do have multiple instances of ActiveMatrix BPM in your installation, each instance of the BPM application must have a unique name. This name identifies that BPM system within the ActiveMatrix enterprise and enables you to be certain when you deploy an application that you are deploying it to the correct version of BPM.

If you have more than one BPM system, you can do either of:

- Amend the default values specified in your deployment server each time you deploy an application,
- Create a separate deployment server for each BPM instance.

When you create a deployment server, you select the default environment name and the default application name within that environment to which you intend to deploy your applications. See [Creating a New Deployment Server](#) for details of this process. When you deploy an application to a deployment server, the **Application Configuration** page is displayed, as described in [Deploying an Application Using the Deployment Wizard](#). As noted in that section, you can specify a different destination environment by from the dropdown in the **Environment Name** field, and you can select a destination folder, from those available in the selected environment, in the **Application Folder** field.



Alternatively you can create a separate deployment server, as described in [Creating a New Deployment Server](#), for each instance of the BPM application. This is more convenient because it prevents you from needing to change the configuration for each deployment.

## Re-Deploying for Live Development a Project that Has Changed

When a set of projects have already been deployed to a server (either using drag-and-drop on to the server in TIBCO Business Studio BPM Edition or using **Deploy Project**), use these steps to re-deploy for Live Development of a project that has changed.

### Procedure

1. Create a Deploy Project (**New > File > Deploy Project**).
2. Select **Add Resources** in the Deploy Project Editor, and select the resources you want to deploy.
3. In the **Administrator Configuration Connection** pane, select the online server to be used (for example, Local Development Server).
4. Select **Re-deploy** in the Deploy Project Editor for all the application resources to be deployed. If only one project has changed and needs to be re-deployed then select the **Re-deploy** action for the project and all of its dependent projects (which will be listed above it in the editor).
5. Select **Configure Application** for all of the changed projects (and also re-generate the DAAs). Any project that changes needs to be re-configured before re-deployment.

6. Click **Generate Scripts** for all of the changed projects.
7. Go to the `Scripts` folder in the Deploy Project.
8. Set `TIBCO-HOME` to the location of the TIBCO Business Studio installation folder.
9. Run the `build.xml` file (right-click and select **Run As > Ant Build**).

## Re-Deploy/Upgrade of a Global Signal Definition Run-time Application

There are a number of issues you should be aware of when re-deploying/upgrading a Global Signal Definition Run-time Application.

Throw / catch events in existing deployed processes will use the latest minor version of the major version that they were originally defined with (for example, a process defined using GlobalSignal 1.0.0, will use GlobalSignal 1.1.0 but not GlobalSignal 2.0).

- A catch signal defined against v1.0.0 will **see** the signal thrown by an event defined against v1.1.0.
- A catch signal defined and against v1.0.0 will **not see** the signal thrown by an event defined against v2.0 of the global signal definition.

The upgrade changes permitted to a global signal definition's payload are restricted to the following:

- Add new optional payload parameter.
- Remove existing optional payload parameter.
- Change a payload parameter (that is, a non-correlation payload parameter) from mandatory to optional. Note, however, if this is done, all processes that catch the global signal must be rebuilt and redeployed, otherwise an exception is thrown at runtime if the parameter value is missing.



- New global signal definitions can be added but existing signal definitions cannot be removed.
- Upgrade to the new major version is possible only as a separate deployment.

Any other changes than those stated above would cause failure of existing deployed throw/catch events.

- Any attempt to upgrade a global signal definition project with the same major version that breaks this rule will result in a failure to deploy.
- You will see an error during deployment (something like `Global signal upgrade validation error: %error details%`) with the details about the first incompatible change encountered while performing validation.

Therefore, because all correlation parameters must be configured as mandatory it means that once the signal has been deployed for the first time, the Signal correlation parameters are **set in stone and cannot be changed unless the major version is changed**.



# Scripted Deployment

TIBCO Business Studio enables you to create deployment scripts. You can assign one or more existing deployable resources to a deployment project, and then use that project to produce build and deployment Apache Ant™ scripts that can be used to deploy multiple projects together, either from TIBCO Business Studio or from the command-line. This process makes it easy to repeatedly deploy the same set of projects during development and testing, or to deploy a set of projects to multiple BPM servers in a production environment.

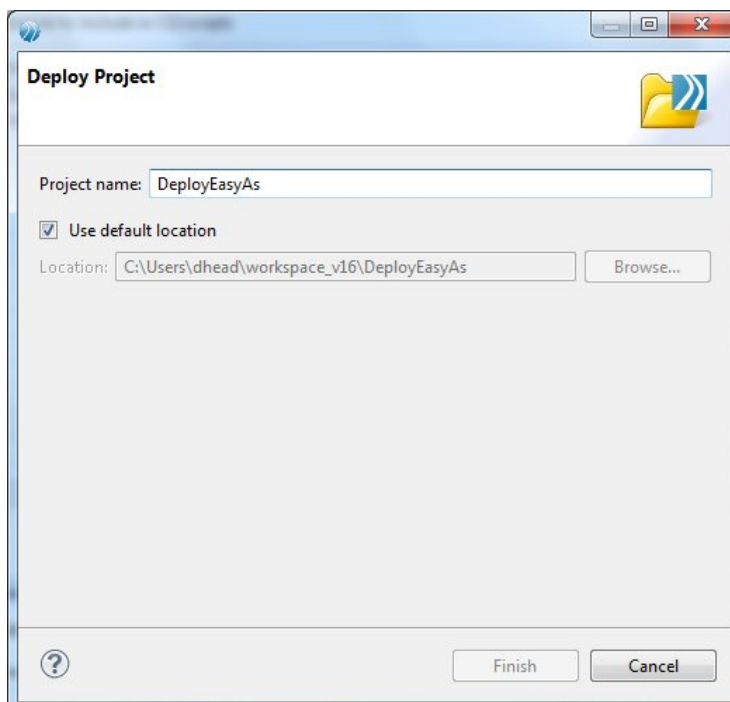


The scripted deployment functionality is provided as a helper tool for users who are familiar with TIBCO Business Studio and TIBCO ActiveMatrix BPM. You are expected to understand the implications of your choices both for the deployment process itself and on existing deployed applications.

## Creating a Deployment Project

### Procedure

1. Select **File > New > Deploy Project**.
2. In the New Deploy Project dialog, enter a **Project name** and click **Finish**.



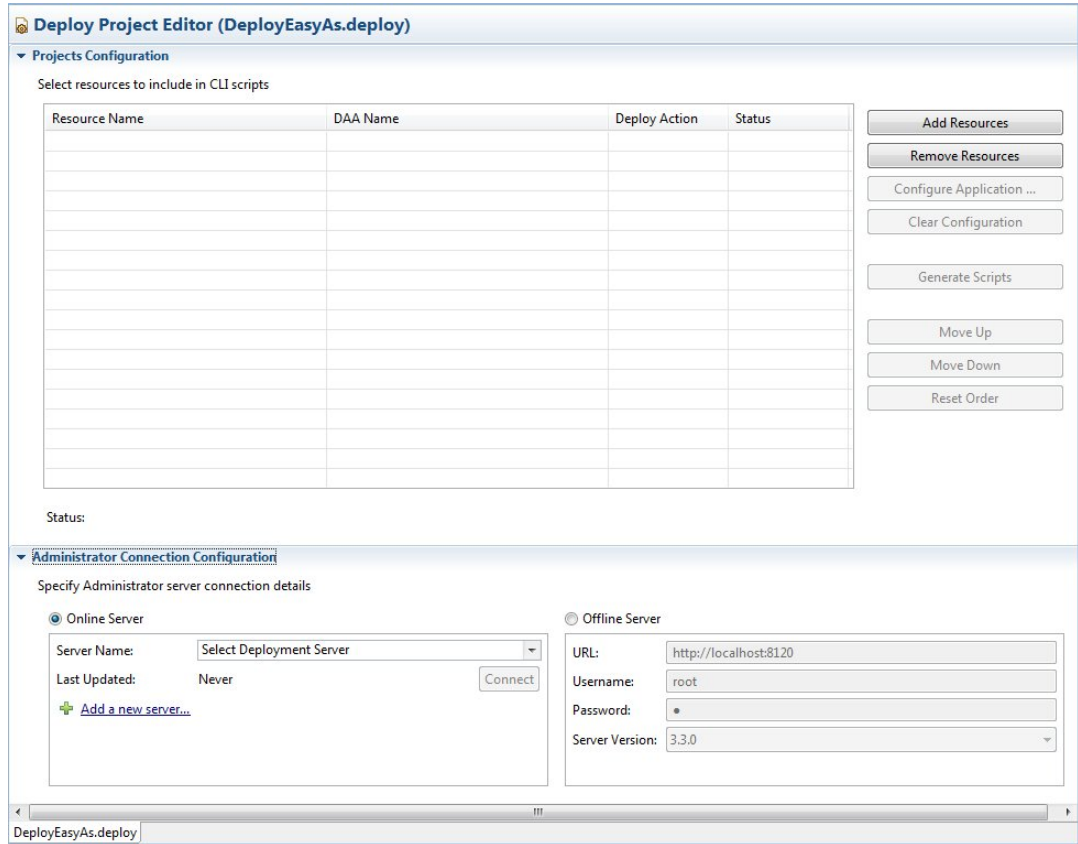
3. The **Deploy Project Editor** for the project displays.

Initially, this editor will be blank. Using this editor, you:

- Specify the server you wish to deploy to,
- Add to the project a number of resources, from which distributed application archives (DAAs) can be created,
- Edit the deployment configuration of each resource,
- Select the action to perform for each resource,
- Set the required deployment and environment configurations for each resource,



- Generate the deployment Ant scripts.
- 

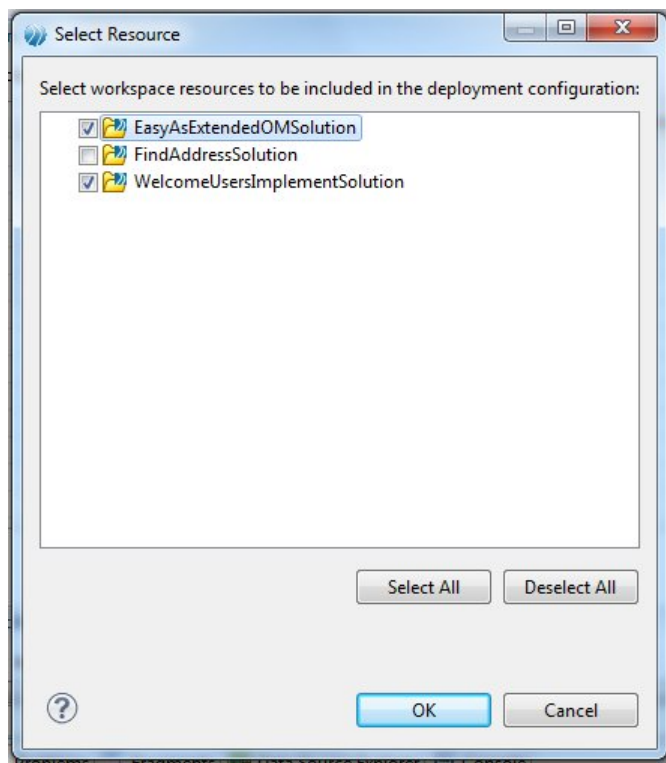


## Adding Resources to the Deployment Project

### Procedure

1. To add an existing project to the deployment project, click **Add Resources**. The Select Resource dialog displays.

A **Resource** is the source of an application DAA. For example, a resource can be a BPM project or a composite of SOA projects.



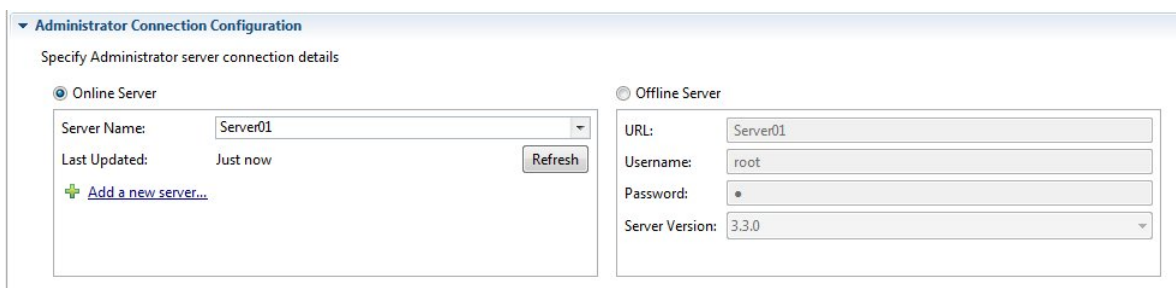
2. Select the projects or other resources you wish to add to the deployment project and click **OK**. The projects and other resources that you selected are now listed in the **Projects Configuration** panel of the **Deploy Project Editor**.

You can use the **Move Up** and **Move Down** buttons in the **Deploy Project Editor** to change the order of the resources within the deployment project, and the **Reset Order** button to undo such changes. The order of resources within the project is the order in which they will be deployed. By default, the resources are ordered based on project dependencies, so that all the resources that a project depends on are listed before that project.

## Specifying the Server

You must identify a deployment server. In the **Administrator Connection Configuration** panel of the **Deploy Project Editor**, specify the deployment server that you intend to use for test deployments.

This server may be currently online or offline. If it is offline, you may specify the server **URL**, **Username**, **Password**, and ActiveMatrix **Server version**.



## Configuring the Resources in the Deployment Project

When you have added the projects and other resources that you want the deployment project to contain, you must configure those resources and generate a DAA for each of the resources. Depending on the state of the BPM projects, you may also need to specify some other details.

This is similar to the deployment process described in [Deploying an Application Using the Deployment Wizard](#).

Prior to configuring the resources, you may need to refresh the deployment server to ensure that its current state is known (for more information, see [Refreshing a Deployment Server](#)).



Also note that if the projects in your Deployment Project are using virtualized bindings, you may need to deploy some the projects first, then configure and deploy the other projects (that are dependent on the earlier deployed projects). You must do it this way so that the virtual bindings are available in the pick list when you are configuring the projects that are dependent on those bindings.

### Procedure

1. For each resource in turn, select it in the **Projects Configuration** panel and select **New** from the **Deploy Action** dropdown (replacing the default **Don't Deploy**).

You can specify **Don't Deploy** in the **Deploy Action** field for resources that you do not want to deploy (this can be useful when you need to deploy a subset of resources that failed initially and have been reconfigured to run the deployment scripts again).

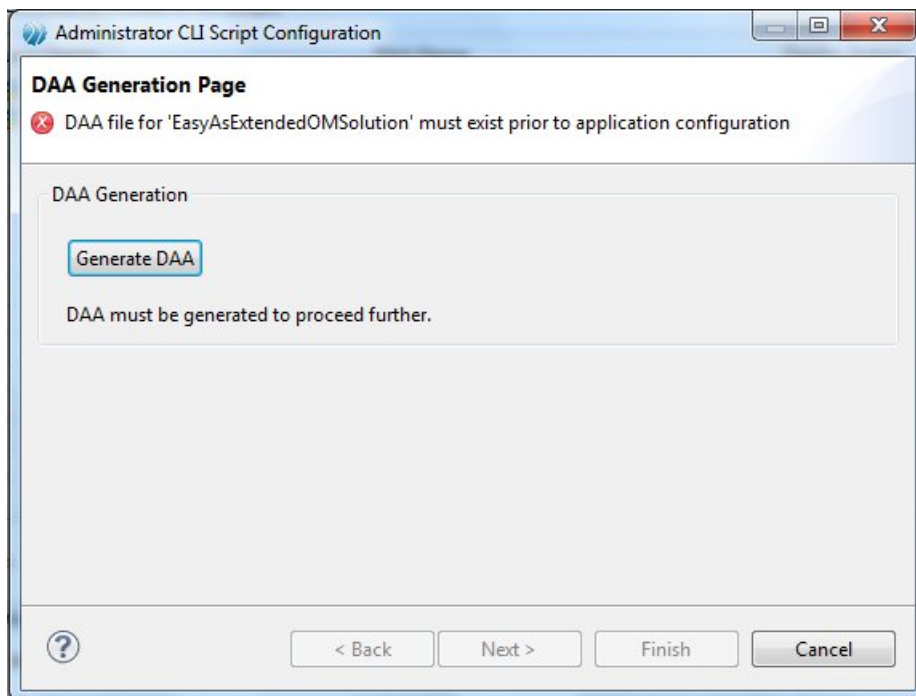
Resource Name	DAA Name	Deploy Action	Status
EasyAsExtendedOMSolution	com.example.easyasextendedomsolution...	Don't Deploy	
WelcomeUsersImplementSolution	com.example.welcomeusersimplementsol...	Don't Deploy	
		New	
		Re-Deploy	
		Upgrade	
		Upgrade & Re-Provision	
		Delete	

2. In the **Status** column, which was initially blank, the status **Not Configured** displays. The **Status** column can have one of the following values:

Status	Description
<blank>	The initial state when a resource is added, or has <b>Don't Deploy</b> specified.
Configured	The resource is configured for deployment.
Not Configured	The resource is not configured for deployment. Click <b>Configure Application</b> to configure it.
Missing Dependency	The resource is missing one or more dependencies. Click <b>Add Resources</b> to select and add the items that this resource depends upon.
Inconsistent Order	The selected resource is out of sequence, that is some of the resources it depends on are listed below it in the table. Use the <b>Move Up</b> , <b>Move Down</b> , or <b>Reset Order</b> buttons to rectify the order.
Not Accessible	The selected resource is not accessible in the workspace. It is either deleted, closed or does not have any deployable artefacts.

Click **Configure Application**.

- The **DAA Generation Page** displays. For scripted deployment to work, a DAA must exist in the deployment project.

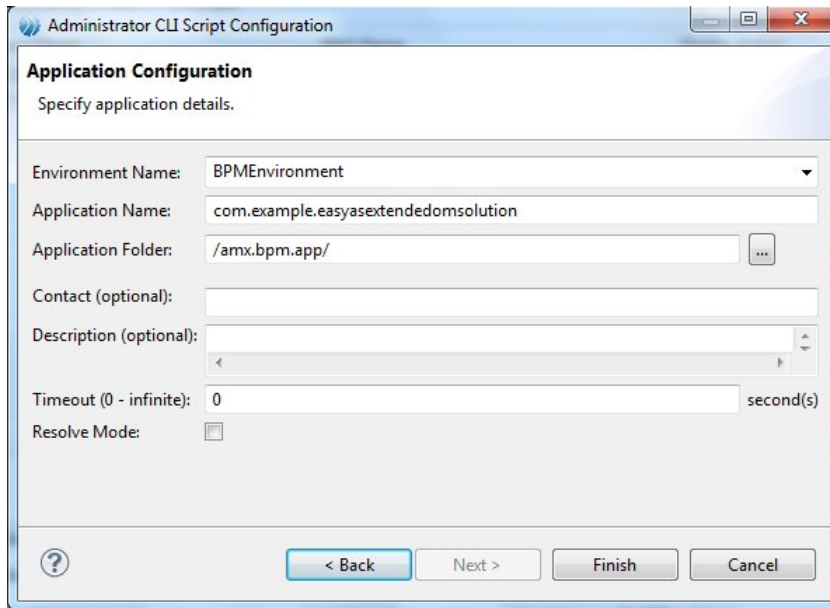


Click **Generate DAA**, and then click **Next**.



If a DAA for this resource already exists within the deployment project, the button on this page shows **Re-Generate DAA** instead. You can ignore this and click **Next** if you are confident that the resource has not changed since the existing DAA was generated, or you can re-generate the DAA now. If the source has changed, you must re-generate the DAA.

- The **Application Configuration** page displays. This shows the default deployment information for the server that you selected in [Specifying the Server](#). This is some of the information that you entered in the **RuntimeServer** Parameters dialog when creating the deployment server - see [Creating a New Deployment Server](#). Amend the **Environment Name**, **Application Folder** and other information displayed if you want to use values different from the defaults that you set up for this deployment server or if the server that you specified is offline.



**Administrator CLI Script Configuration**

**Application Configuration**  
Specify application details.

Environment Name: BPMEnvironment

Application Name: com.example.easyasextendedomsolution

Application Folder: /amx.bpm.app/

Contact (optional):

Description (optional):

Timeout (0 - infinite): 0 second(s)

Resolve Mode:

< Back Next > Finish Cancel

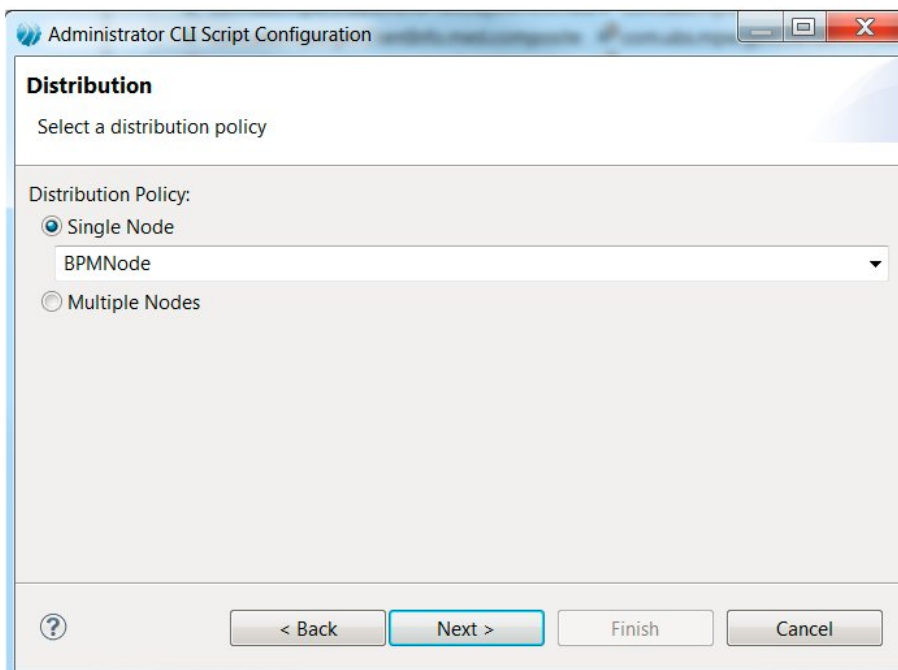
5. At this point you may be able to click **Finish**. However if the resource needs any further configuration, the **Finish** button will not be available. You should click **Next** and complete any further dialogs that display.
6. When you click **Finish**, the value in the **Status** column changes to **Configured**. Repeat the configuration for each resource in the deployment project.

## Result

### Distribution

The **Distribution** page may then be displayed. Select whether you wish the resources to be deployed to one node or more, and then select the node from the drop downs provided.

Click **Next**.



**Administrator CLI Script Configuration**

**Distribution**  
Select a distribution policy

Distribution Policy:

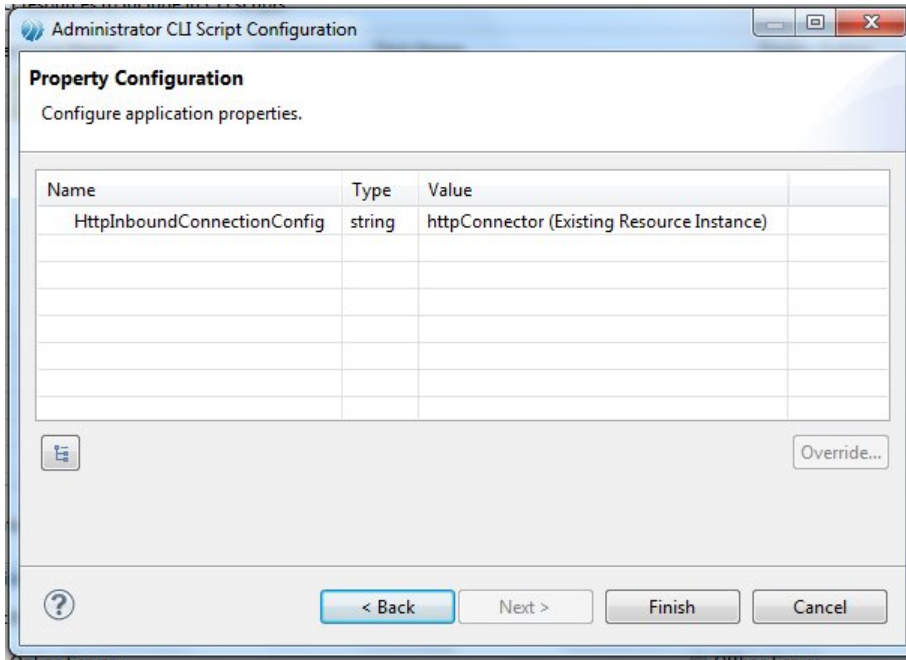
Single Node

BPMNode

Multiple Nodes

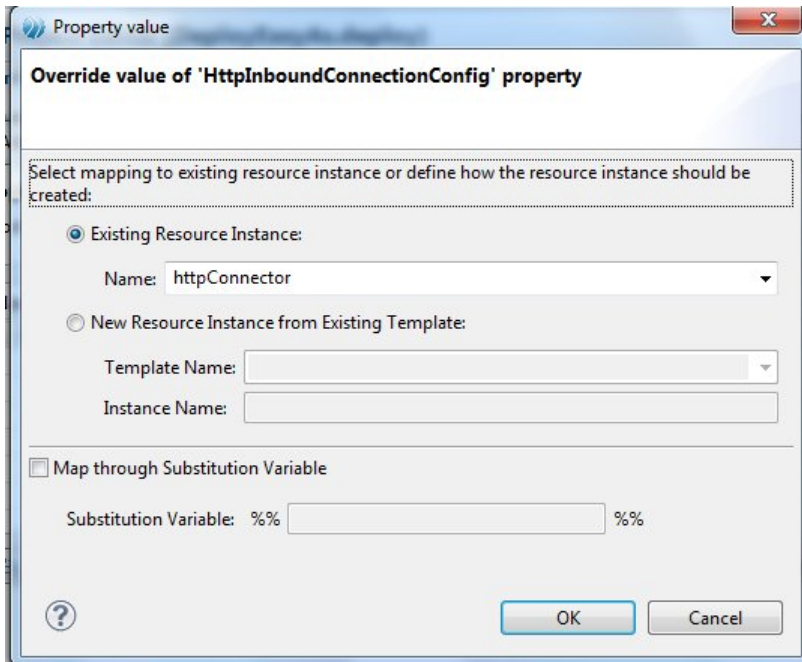
? < Back Next > Finish Cancel

The **Property Configuration** page is displayed if any property mappings are required.



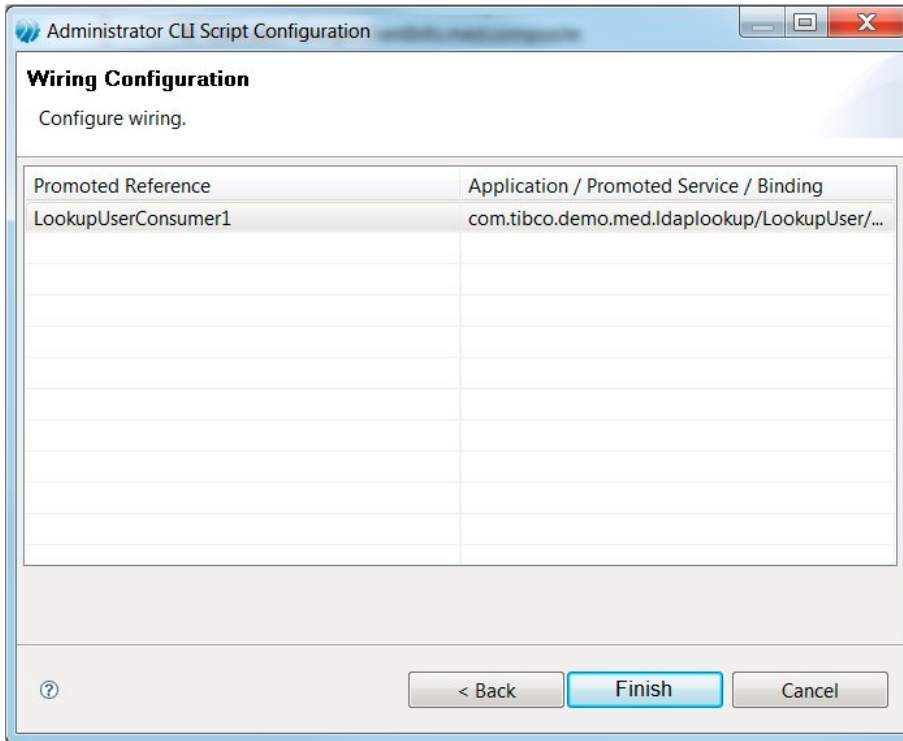
Select a property and click **Override** to change the specified value assigned to any of the properties in the **Name** column. The **Property value** page displays.

By default, the properties are mapped to substitution variables, which provide a level of indirection. To change this, specify the new value in the Property value dialog and click **OK**.

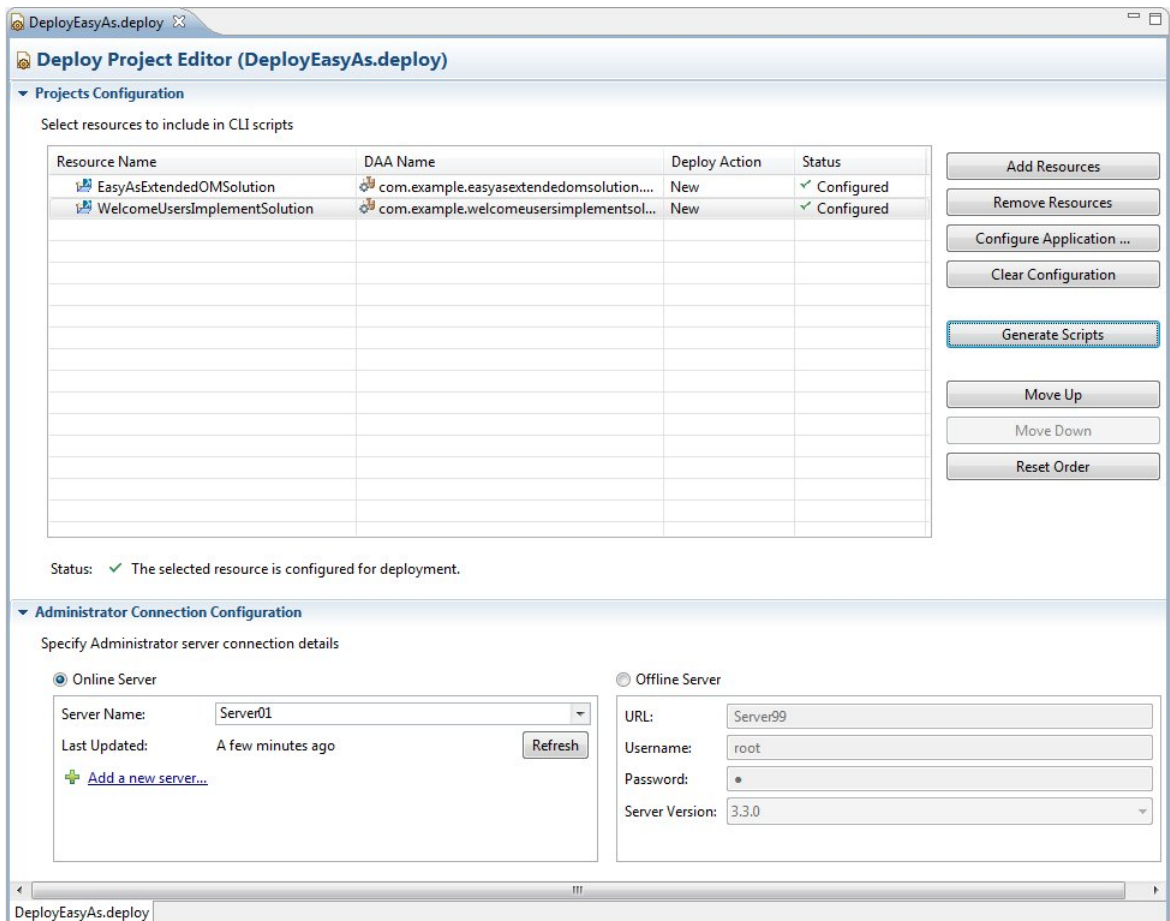


The **Wiring Configuration** page is displayed if the resource contains unresolved promoted references.

1. Click **-add wire-** in the row containing the unresolved promoted reference. A drop-down listing the available applications, services, and bindings appears.
2. Select the application, service or binding that you want to wire the promoted reference with, and click outside the text
3. When you have done this for all the listed references, click **Finish**.



The **Deploy Project Editor** for the project is now populated.





## Generating the Deployment Scripts

When you have configured the resources in the deployment project, you generate the deployment script.

### Procedure

1. Ensure that every resource that you want to include in the deployment script has a **Deploy Action** of:

- **New**, if it is to be deployed for the first time,
- **Redeploy, Upgrade, or Reprovision** if it has been deployed previously.

Ensure that every resource that you want to include has a **Status** of **Configured**. Any resources that you do not need to include must have the **Deploy Action** set to **Don't Deploy**.

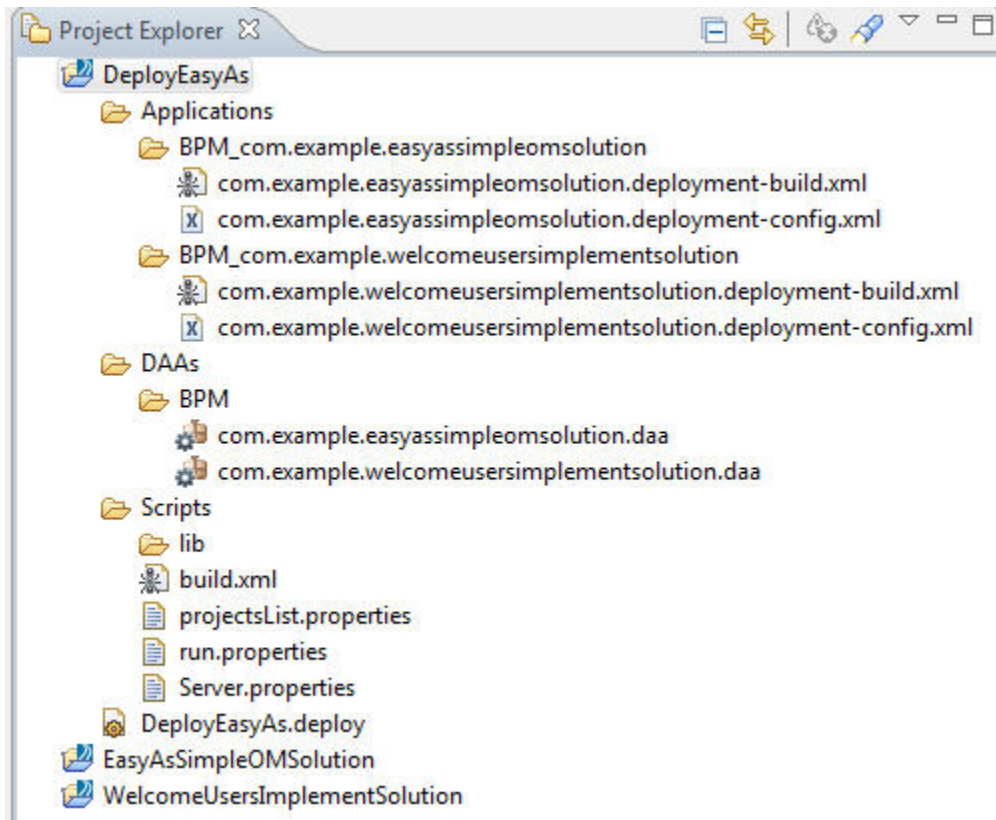


The **Generate Scripts** button is only enabled when all the projects to be deployed (that is, not having **Don't Deploy** action) are **Configured**. If you change any configuration details, you must regenerate the scripts.

2. Save the project.
3. Click **Generate Scripts**.

### Result

Once the scripts have been generated, they are visible in the file structure of the deployment project in Project Explorer.



In this file structure:





You should not normally need to edit any of the files in this file structure, with the exception of **run.properties** and **server.properties**.

Note that the generated files (including **run.properties** and **server.properties**) are overwritten when the **Generate Scripts** button is clicked.

- The **Applications** folder contains sub-folders for each application, named **BPM\_***daaname* or **SOA\_***daaname*, depending on the type of project. For each project:
  - *daaname.deployment-build.xml* contains the tasks based on the value selected for **Deploy Action** for the resource. The Ant target defaults to **all**.
  - *daaname.deployment-config.xml* contains the resource's configuration details.
- The **DAAs** folder contains the generated DAAs.
- The **Scripts** folder contains the generated scripts.
  - **build.xml** is the build file which you will subsequently run, from the command line, to deploy the projects. The default Ant target **run** must be used. The **run** target invokes the target defined in the **run.properties** file ( **deploy** or **import-and-deploy**). The **deploy** target invokes the default target of an application as specified in its generated build file *daaName.deployment-build.xml*.
  - The log files **error.log** and **report.log** are generated in this folder when **build.xml** is run. The **error.log** file is where error details are written if script execution fails. The **report.log** file shows a summary of the script execution, in terms of actions performed and remaining actions in case of failure.

```

----- Script Execution Summary -----

2013-08-13 16:12:33.353
Total Resources:      2
----- Performed Actions -----

- Deploy: EasyAsSimpleOMSolution
- Deploy: WelcomeUsersImplementSolution

----- Unperformed Actions -----

*See log file 'C:\temp\workspace\.metadata/.log' for further details.

```

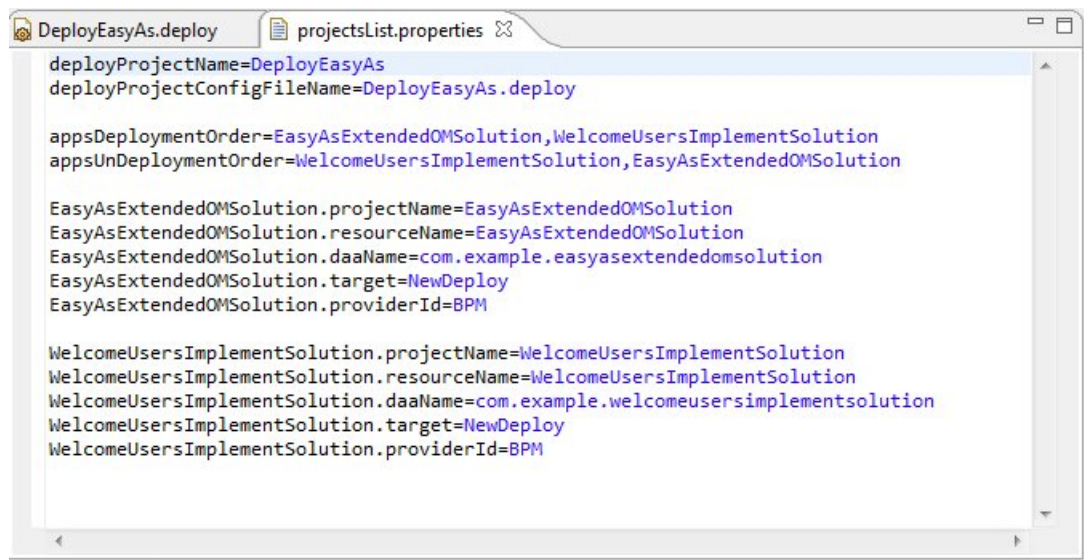
- **run.properties** contains the properties to configure the **build.xml** file, including the task to execute (**deploy** or **import-and-deploy**), workspace locations, **TIBCO\_HOME** location, **import.locations**, checkout script and location. Comments in the file explain what to set.



You can edit this file to specify the **import-and-deploy** target for the **build.xml** file when deploying the project.

When you specify a location in **run.properties**, use either '\\\ ' or '/' to introduce the pathname.

- **server.properties** contains the properties for the selected deployment server including **adminURL**, **username**, **password**, **httpConnection**, and **Timeout**.
- **projectsList.properties** lists the resources in the deployment project and defines their properties.



```

deployProjectName=DeployEasyAs
deployProjectConfigFileName=DeployEasyAs.deploy

appsDeploymentOrder=EasyAsExtendedOMSolution,WelcomeUsersImplementSolution
appsUnDeploymentOrder=WelcomeUsersImplementSolution,EasyAsExtendedOMSolution

EasyAsExtendedOMSolution.projectName=EasyAsExtendedOMSolution
EasyAsExtendedOMSolution.resourceName=EasyAsExtendedOMSolution
EasyAsExtendedOMSolution.daaName=com.example.easyasextendedomsolution
EasyAsExtendedOMSolution.target=NewDeploy
EasyAsExtendedOMSolution.providerId=BPM

WelcomeUsersImplementSolution.projectName=WelcomeUsersImplementSolution
WelcomeUsersImplementSolution.resourceName=WelcomeUsersImplementSolution
WelcomeUsersImplementSolution.daaName=com.example.welcomeusersimplementsolution
WelcomeUsersImplementSolution.target=NewDeploy
WelcomeUsersImplementSolution.providerId=BPM

```

- The *projectname.deploy* file includes user selections and configuration details.



If this **.deploy** file is deleted, you will need to regenerate the project.

- The **lib** folder inside the **Scripts** folder contains:
  - the Ant jar file
  - the `log4j.properties` file

These files are used by Ant during the execution of generated scripts.

## Modifying a Deployment Project

You can modify an existing deployment project to include different resources or to reconfigure the resources.



Remember that you need to click **Generate Scripts** before executing the scripts if anything in the deployment project has changed.

## Adding a Resource

You can add resources to a deployment project by clicking **Add Resources** in the **Deploy Project Editor**.

Configure the added resource in the same way as when creating a new deployment project, and re-generate the scripts.

## Removing a Resource

You can remove a resource from a deployment project by clicking **Remove Resources** in the **Deploy Project Editor**. You are asked to confirm the deletion, and then the resource, its generated DAA, and all its configuration files are removed.

Other files are not modified, so you need to re-generate the scripts.

## Reconfiguring a Resource

Reconfigure a resource if one of the resources has changed, for example if a project has been altered after testing.

### Procedure

1. Select the affected resource in the **Deploy Project Editor**.
2. Click **Clear Configuration** and confirm when prompted.
3. Click **Configure Application** and repeat the configuration process to generate a new DAA.
4. Re-generate the scripts.

## Changing the Deployment Order of Resources

You can use the **Move Up** and **Move Down** buttons to change the order of the resources within the deployment project, and the **Reset Order** button to undo such changes. You do not need to reconfigure moved resources, but do need to re-generate the scripts.

The order of resources within the project is the order in which they will be deployed. By default, the resources are ordered based on project dependencies, so that all the resources that a project depends on are listed before that project. If you change this order in such a way as to invalidate a dependency, an **Inconsistent order** warning message is shown in the **Status** column. Deploying without correcting the order may cause the generated scripts to fail.

## Executing the Deployment Script

You can execute the deployment project scripts either from the same computer on which you created it, or on a different computer which must have TIBCO Business Studio installed. Scripts can be run from TIBCO Business Studio or from a command line.

- Before executing the scripts as outlined below, make sure to update the `run.properties` file as explained in: [Generating the Deployment Scripts](#).
- Running scripts using the command-line:
  - You need to have installed **Apache ant** (from <http://ant.apache.org>).
  - Then go to the `Scripts` folder under the deploy project in a command shell and use this command: `%ANT_HOME%/bin/ant -f build.xml`. Note that `%ANT_HOME%` is the folder location where Ant is installed.
- Running scripts using TIBCO Business Studio:
  - In order to run the scripts from TIBCO Business Studio workbench, go to the `Scripts` folder inside the deploy project and right-click on `build.xml` and select **Run As > Ant Build**.

## Checking in and out of a Repository

You can save the deployment project to a source control repository in order to reuse the generated resources for repeated deployment of selected projects without configuring each time.

See "Source Control" in *TIBCO Business Studio Modeling User's Guide* for information on using the Subversion source repository with TIBCO Business Studio.

You can then:

- Check the deployment project out of the repository to a disk location on the computer from which you intend to deploy the projects.

- Optionally create a checkout script, which can be used by the **import-and-deploy** command to check out the projects to be deployed directly from the repository location.

## Deploying the Project

### Procedure

- Edit the `run.properties` file to specify the configuration properties for the `build.xml` file.

```
# Copyright (c) 2013 TIBCO Software Inc.
# All Rights Reserved.
#
# This software is the confidential and proprietary information of TIBCO Software Inc.

# Set default target to be executed (deploy or import-and-deploy)
default.target=import-and-deploy

#Note: use either '\\' or '/' when specifying a path.

# Set this to studio installation directory
TIBCO-HOME=C:\\apps\\TIBCO\\studio-bpm-3.6

# Set this temporary location, which will be used as a workspace by the studio during the scripts execution.
# The contents of this directory will be deleted automatically before the scripts are executed.
temp.workspace=C:\\temp\\workspace

#Note: Either set 'import.location' or 'checkout.file' below. If both are set then 'import.location' will be used
# instead of 'checkout.file'.

# Set this to import location directory from where the projects will be imported into 'temp.workspace' and are
# deployed after the DAAs generation.
import.location=C:\\temp\\importDir

# Set the location of checkout script ant file to get projects from a repository and also provide the
# 'checkout.location' where the projects will be checked out by the checkout file.
checkout.file=C:\\temp\\checkout.xml
checkout.location=C:\\temp\\checkout

#set this to 'true' to import projects exported as archive files
use.archives=false

ADMIN-VERSION=3.3

#set this to 'true' for verbose output of the scripts execution
verbose=false

#####

# Advanced options
# Avoid changing the properties below if not sure

#####
projectsList.file=projectsList.properties
```

- Set **default.target** to **import-and-deploy**.
  - Set **import.location** to the location from which the scripts will be imported.
  - Set **use.archives** to **true** if the projects being imported are archive files.
- From a command line, run the `build.xml` file to deploy the projects on a test server. This file takes the target information it needs from `run.properties` and `server.properties`, so you do not need to edit `build.xml` directly.



Whenever you run the `build.xml` file, the log files `error.log` and `report.log` are generated, in the same folder as `build.xml`.

The **import-and-deploy** command imports the projects, rebuilds them, regenerates the DAAs and the relevant scripts, and deploys the projects to the test deployment server. The command also imports the deployment project itself to **temp.workspace** and the DAAs and scripts are generated there. Note that the **temp.workspace** directory is cleared before importing the projects.

3. If you are working in a production environment, once you have successfully deployed your projects on the test server ([step 2](#)), you can update the `server.properties` file to replace the details of your test server with those of the live server.
4. Edit the `run.properties` file to change the target to **deploy**.

```
# Set default target to be executed (deploy or import-and-deploy)
default.target=deploy
```

5. Run `build.xml` again. With the target now set to **deploy**, this uses the DAAs generated in [step 2](#) and the updated server information to deploy the projects on to the live server.

## Sample Projects - Shared Resource Create Install and Subversion (SVN)

Sample projects are supplied with TIBCO Business Studio to allow you to try out the creation and installation of shared resources in your own environment, and also integrating with Subversion to collect files.

The sample files are located here:

```
TIBCO_HOME\tibco\<environment location>\studio\n.n\samples\WS01.zip
```

## Creating Installing Shared Resources

### Procedure

1. Change `TIBCO_HOME` in both build files to the appropriate location.
2. In `Server.properties`, supply the correct IP address for your own settings.
3. Run the two build files. They create a required resource instance and template.

## Using Subversion

### Procedure

1. Edit `run.properties`. The file contains two options for setting the location where projects will be imported from:
  - `import.location` (non-SVN)
  - `checkout.location` (SVN option): you need to define the checkout file (an ant project)
2. In `build.properties` you need to define values for the following:
  - `SVN.url`
  - `SVN.username`
  - `SVN.password`.

**Work.dir** is the same as **checkout.location** (in the `run.properties` file)

Checkout location is:

```
work.dir=C:\\temp\\checkout
```



# Undeployment

This section provides guidelines on undeploying applications that have been deployed to your BPM runtime.

You can undeploy a deployed application by doing one of the following:

- By removing it directly using TIBCO Business Studio. This is a suitable method where projects need to be deployed and undeployed rapidly for development or testing purposes. It may not be suitable in a production environment, where developers are unlikely to have direct access to production servers. See [Undeploying an Application Directly from TIBCO Business Studio](#) for details.
- Using the Administrator interface in the BPM runtime environment. This is the method that would typically be used in a production environment. See [Undeploying an Application Using Administrator](#) for details of how to do this, and consult the Administrator interface documentation for your BPM runtime environment if you need any further information.
- By running a deployment script. See [Undeploying an Application Using a Deployment Script](#).

## Undeploying an Application Directly from TIBCO Business Studio

### Procedure

1. In the Deployment Servers view, right-click on the deployed project and select **Remove Application....**



Note that there is no "undeploy" option in TIBCO Business Studio.

2. A progress meter is briefly displayed in the status bar. TIBCO Business Studio, it attempts a graceful undeployment of the application. If there are no outstanding open work items and/or process instances, the application is removed from the list of deployed applications.

However, graceful undeployment can be vetoed by the BPM runtime. For example, the BPM runtime will not undeploy the application if it has outstanding open work items and/or process instances. In this case, TIBCO Business Studio waits for a graceful undeploy to complete and does not attempt a forceful undeploy.



Make sure that you have successfully undeployed a BPM application before trying to Delete it. Do not use the **Force Delete** option to remove a BPM application, except in the situations described in [Cleaning Up Undeployed Applications](#). Doing so removes artifacts only from Administrator, and leaves Process Manager and other TIBCO ActiveMatrix BPM components out of sync, which can cause problems on restart.

## Undeploying an Application Using Administrator

You can delete a deployed application by using the Administrator GUI in TIBCO ActiveMatrix Administrator. The application must be running for the undeploy to complete. However, you can always initiate an undeploy and the system will queue it and apply it to components as they become available. During undeployment, dependencies are taken into account to allow processing to clean up before removing components and bindings.



Do not perform the Undeploy nor Force Undeploy operation on the BPM product application, which is named "amx.bpm.app" by default. Performing either of those operations on the BPM product application can cause the system to become unstable and/or unrecoverable, possibly requiring a re-creation of the environment.

## Procedure

1. Select **Applications**.
2. In the Applications list, click one or more applications.
3. From the **Undeploy** drop-down, select one of the following options:

## Result

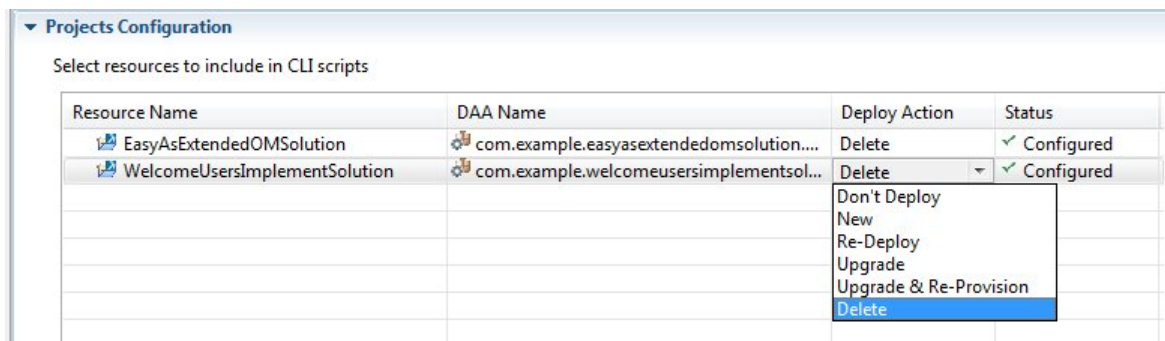
Option	Procedure
Undeploy This option checks dependencies on the selected applications.	<p>If any of the selected applications has dependencies, the Application Dependencies to Undeploy dialog displays with target applications. Check the checkboxes next to the selected applications to continue with undeployment, and click <b>Undeploy</b>.</p> <p>The selected applications are undeployed. The length of time this action takes to complete depends on how long it takes for the selected applications to complete their processing.</p>
Force Undeploy This option does not check dependencies on the selected applications.	<p>A dialog asks you to confirm the forced undeployment. Components in the selected applications are allowed to perform cleanup operations.</p> <p>If a component of the application stores information in a database or file, that data may remain after a force undeploy and must be cleaned up manually.</p>

## Undeploying an Application Using a Deployment Script

You can undeploy an application using the scripted deployment techniques. See [Scripted Deployment](#).

## Procedure

1. In the **Deploy Project Editor** for an existing deployment project, select the resources that you want to undeploy. For each one, set the **Deploy Action** to **Delete**.



2. Regenerate the deployment scripts, as described in [Generating the Deployment Scripts](#).
3. Check the value in the **run.properties** file. Set **default.target** either to **deploy** or to **import-and-deploy**, depending on whether or not you are importing your deployment scripts from a source control repository. See [Deploying the Project](#) for further details.
4. From a command line, run the **build.xml** file to deploy the projects on a test server, as described in [Deploying the Project](#).



The **deploy** or **import-and-deploy** command performs the deployment action specified for each resource. If you have specified this as **Delete**, as in [step 1](#) above, this will be to delete the project from the BPM server.

## Tips for Successful Undeployment

This section lists the points that are best practice for undeploying applications.

- Do not attempt to undeploy applications with outstanding process instances or work items. If an attempt to undeploy an application stops at '**Preparing to Undeploy**', you should find and cancel any outstanding process instances or work items for that application.

If you cancel a process instance that still has an open work item (for example the work item's status is OPENED), then the work item becomes hidden, and the application will not undeploy. You can check there are no hidden work items:

- In Workspace, by selecting either **Only show hidden work items** or **Show visible and hidden work items** from the **View** menu on the work item list.
- In Openspace, by selecting **Visible Equal FALSE** from the Configure Filter dialog.

You need to be sure that you are logged in as a user who has the correct privileges to see work items for this process.

- If an attempt to undeploy an application does not remove all its components, you can use the cleanup utilities available from TIBCO Support. See [Cleaning Up Undeployed Applications](#).



Do not perform the Undeploy nor Force Undeploy operation on the BPM product application, which is named "anx.bpm.app" by default. Performing either of those operations on the BPM product application can cause the system to become unstable and/or unrecoverable, possibly requiring a re-creation of the environment.

## Cleaning Up Undeployed Applications

If you try to undeploy, or forced undeploy, an application using either the Administrator UI or the CLI, and the undeploy process does not remove all the components from the node runtime, cleanup utilities are available from TIBCO Support.



The cleanup utilities remove all versions of an application. In other words, you cannot use the cleanup utilities to remove a later version of an application, leaving an earlier version present. This is not supported.



- You should only use the cleanup utilities with advice from TIBCO Support.
- You must obtain the latest version of `nodeutil` from TIBCO Support.

This section describes using both:

- the BPM database cleanup utility `bpm-db-cleanup`, and
- the ActiveMatrix node cleanup utility `nodeutil`

to remove any components that the undeploy has failed to remove. A typical sequence for using these utilities might be:

### Procedure

1. Undeploy or Force Undeploy an application using Administrator UI or CLI. This step should remove application components from a Node's runtime.
2. Delete an application using Administrator UI or CLI. This step removes information about the application and its components from the Administrator's database.

3. If either of the above steps fail, grant yourself, or the username that will be used, the permissions to perform a forced delete.

The Force Delete option is only visible, and Force Delete enabled, if you have the necessary permissions. By default no user has the necessary permissions to use Force Delete. See the section "Permissions" in *TIBCO ActiveMatrix Administration* for information on setting these permissions in Administrator.

4. Use Force Delete using the Administrator UI or CLI to remove information about the application and its components from the Administrator's database
5. Use the BPM database cleanup utility `bpm-db-cleanup` to remove rows from the various database tables that were created when the process was originally deployed. See "The bpm-db-cleanup Utility" in *TIBCO ActiveMatrix BPM Troubleshooting* for these utilities..



TIBCO recommends that you disable the Force Delete option once you have finished this procedure, by removing in Administrator the permissions granted in [step 3](#).

6. Use the ActiveMatrix node cleanup utility **nodeutil** to remove from the Node's runtime any components of the application that were not cleared by the previous steps.

# Application Upgrade

This section explains how to upgrade deployed applications.

A process can expose a web service operation that other processes or applications can invoke. In this case, the process acts as the service provider in the conversation, and a WSDL defines the service interface to that operation. The WSDL can be either imported or automatically generated by TIBCO Business Studio.

Once the application containing the process has been deployed to the BPM runtime, it can only be subsequently upgraded if its service interface (defined by the WSDLs used to expose its services) has not changed. (See [Deploying the Organization Model](#).)

If a process uses a **generated** WSDL, making changes to the project may result in changes to the generated WSDL, meaning that the application cannot be upgraded.

See "WSDL Change Considerations for Application Upgrade" in *TIBCO Business Studio - BPM Implementation* for information on how best to ensure that changes to your applications do not lead to any incompatibility that will prevent you easily upgrading them.

Note that the **project** is the unit of deployment, so you cannot deploy a subset of a project. This means that if a sub-process or a package within a project has been upgraded, you cannot redeploy that element on its own but must redeploy the entire project.

## Upgrading a Deployed Application

You can upgrade a previously deployed application using the same methods used to deploy it in the first place.

See [Deploying an Application from TIBCO Business Studio](#).

### Procedure

1. Make sure that no changes have been made to the application that would prevent upgrade. After an application has been initially deployed to the BPM runtime, it can only be subsequently upgraded if its service interface (defined by the WSDLs used to expose its services) has not changed.
2. Make sure that the application has exactly the same name as the one that you wish to upgrade. See [Deployed Applications](#).
3. Deploy the application by doing one of the following:
  - Using a deployment server. See [Upgrading an Application Using a Deployment Server](#).
  - Exporting to a Distributed Application Archive (DAA). See [Exporting to DAA](#).
  - Running a deployment script. See [Upgrading an Application Using a Deployment Script](#).

## Reverting to the Original Version of an Upgraded Application

If you encounter a problem during deployment or operation of the upgraded application, you may subsequently want to revert to the original version of the application (by force undeploying the new version from ActiveMatrix Administrator).

However, a WSDL validation error will occur when you try to do this if changes made to the WSDL, though valid for upgrade, are detected as a change to the service interface when attempting to downgrade. (ActiveMatrix Administrator does not distinguish at this level between upgrade and downgrade, so this problem can occur even though the version of the WSDL being downgraded may never have been used, and even though the version being downgraded to will be perfectly compatible with the version of the application that will be active.)

For example, if you add a new operation this is valid when upgrading - but when downgrading you will be removing that operation. This will be detected as a change to the service interface and so is not permitted.

If this happens, you can force ActiveMatrix Administrator to skip WSDL validation and so allow you to revert this application to the original version. See "Troubleshooting > Applications > Unable to revert to older version of an application" in *TIBCO ActiveMatrix BPM SOA Administration* for more information about how to do this.

## Upgrading an Application Using a Deployment Server

If you are using a deployment server, the **Upgrade Existing Application** checkbox on the Administrator Setup dialog of the **DAA Deployment Wizard** is enabled. Select or clear this checkbox to choose the desired application upgrade behavior, as shown in the following table. (By default, the checkbox is selected).

See **Administrator Setup**.

Upgrade Existing Application	Behavior
Unselected	<ul style="list-style-type: none"> <li>• TIBCO Business Studio attempts a graceful undeployment of any application with the same name and waits for all the outstanding work items or process instances for the application to be completed.</li> <li>• The application state is updated to Preparing for Undeploy.</li> <li>• Graceful undeployment can be vetoed by the BPM runtime if there are outstanding work items or process instances.</li> <li>• In order to complete the deployment process, you must ensure that all the outstanding work items and process instances for the existing version of application are completed.</li> </ul>
Selected	<ul style="list-style-type: none"> <li>• The application is deployed immediately and does not wait for any outstanding work items or process instances to be completed.</li> <li>• A new application template is created and used for any new work items or process instances created. However, outstanding work items or process instances continue to use the old application template.</li> <li>• When you upgrade an application template, a new application template version is deployed. However, the old application template is not deleted. If needed, you can configure your application to use the old application template from the Administrator UI.</li> </ul>

## Upgrading an Application Using a Deployment Script

You can upgrade and existing application using the scripted deployment techniques.





See [Scripted Deployment](#).

### Procedure

1. In the **Deploy Project Editor** for an existing deployment project, select the resources that you want to upgrade. For each one, set the **Deploy Action** to **Upgrade**.

▼ Projects Configuration

Select resources to include in CLI scripts

Resource Name	DAA Name	Deploy Action	Status
 EasyAsExtendedOMSolution	 com.example.easyasextendedomsolution....	Upgrade	✓ Configured
 WelcomeUsersImplementSolution	 com.example.welcomeusersimplementsol...	Upgrade	✓ Configured
		<ul style="list-style-type: none"> <li>Don't Deploy</li> <li>New</li> <li>Re-Deploy</li> <li>Upgrade</li> <li>Upgrade &amp; Re-Provision</li> <li>Delete</li> </ul>	

2. Regenerate the deployment scripts, as described in [Generating the Deployment Scripts](#).
3. Check the value in the **run.properties** file. Set **default.target** either to **deploy** or to **import-and-deploy**, depending on whether or not you are importing your deployment scripts from a source control repository. See [Deploying the Project](#) for further details.
4. From a command line, run the **build.xml** file to deploy the projects on a test server, as described in [Deploying the Project](#).

The **deploy** or **import-and-deploy** command performs the deployment action specified for each resource. If you have specified this as **Upgrade**, as in [step 1](#) above, this will be to delete the old version of the application from the BPM server and replace it with the upgraded version.

## Deploying Language Packs

---

TIBCO publishes language packs on the TIBCO Software Product Download website for a range of languages. You can deploy these to either TIBCO Openspace or TIBCO Workspace.

From ActiveMatrix BPM 3.0.0 onward, TIBCO is starting to publish language packs as Distributed Application Archives (DAAs). You should use DAA language packs instead of the former Multilingual User Interface (MUI) language packs. DAA language packs can be deployed by ActiveMatrix Administrator, in the same way as any other DAA.

For some languages, Multilingual User Interface (MUI) language packs will continue to be available until they are replaced later by DAA language packs. MUI language packs can be deployed using the documentation supplied with the language pack.

# TIBCO Documentation and Support Services

---

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The website is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The following documentation for TIBCO BPM Enterprise is available on the [TIBCO BPM Enterprise Product Documentation](#) page:

- TIBCO BPM Enterprise Release Notes
- TIBCO BPM Enterprise SOA Concepts
- TIBCO BPM Enterprise Concepts
- TIBCO BPM Enterprise Developer's Guide
- TIBCO BPM Enterprise Web Client Developer's Guide
- TIBCO BPM Enterprise Tutorials
- TIBCO BPM Enterprise Business Data Services Developer Guide
- TIBCO BPM Enterprise Case Data User Guide
- TIBCO BPM Enterprise Event Collector Schema Reference
- TIBCO BPM Enterprise - Integration with Content Management Systems
- TIBCO BPM Enterprise SOA Composite Development
- TIBCO BPM Enterprise Java Component Development
- TIBCO BPM Enterprise Mediation Component Development
- TIBCO BPM Enterprise Mediation API Reference
- TIBCO BPM Enterprise WebApp Component Development
- TIBCO BPM Enterprise Administration
- TIBCO BPM Enterprise Performance Tuning Guide
- TIBCO BPM Enterprise SOA Administration
- TIBCO BPM Enterprise SOA Administration Tutorials
- TIBCO BPM Enterprise SOA Development Tutorials
- TIBCO BPM Enterprise Client Application Management Guide
- TIBCO BPM Enterprise Client Application Developer's Guide
- TIBCO Openspace User's Guide
- TIBCO Openspace Customization Guide
- TIBCO BPM Enterprise Organization Browser User's Guide (Openspace)
- TIBCO BPM Enterprise Organization Browser User's Guide (Workspace)
- TIBCO BPM Enterprise Spotfire Visualizations
- TIBCO Workspace User's Guide



- TIBCO Workspace Configuration and Customization
- TIBCO Workspace Components Developer Guide
- TIBCO BPM Enterprise Troubleshooting Guide
- TIBCO BPM Enterprise Deployment
- TIBCO BPM Enterprise Hawk Plug-in User's Guide
- TIBCO BPM Enterprise Installation: Developer Server
- TIBCO BPM Enterprise Installation and Configuration
- TIBCO BPM Enterprise Log Viewer
- TIBCO BPM Enterprise Single Sign-On

### **How to Contact TIBCO Support**

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

### **How to Join TIBCO Community**

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

## Legal and Third-Party Notices

---

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, TIBCO Administrator, Business Studio, Enterprise Message Service, Hawk, iProcess, JasperReports, and Spotfire are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme.txt file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2005-2022. TIBCO Software Inc. All Rights Reserved.