# Table of Contents

## Preface
- About this documentation
- User types
- Information on documentation
- Conventions used in this manual
- Product support contact information

## Chapter 1: OpenEdge Business Process Server overview
- Business Process Server components
- How Business Process Server works
- Business Process Server user types

## Chapter 2: Setting Password Security and Encryption
- Password security framework
  - Password security framework interfaces
  - Password security policies supported
  - Password validation
- Enabling password security in Business Process Server
- Password security default implementation
  - Default handlers
  - Default password policies configuration
  - Change at first login

## Chapter 3: Configuring File-based Document Management System (DMS)

## Chapter 4: Using the Archiver
- Archiver architecture
- Data handlers
  - Data handlers dependency
  - Data extractor handler
  - Data purging handler
  - Data FileWriter handler
- Archiver scheduler
  - Quartz scheduler configuration
  - Archive scheduler configuration
## Contents

Persistence of archiver jobs ......................................................................................................42  
Scheduler startup and shutdown.............................................................................................42  
Archiver client..........................................................................................................................43  
Archiver configuration..............................................................................................................44  
BMServers tag..........................................................................................................................45  
DataHandlers tag.....................................................................................................................45  
FileStore and DBStore tags......................................................................................................45  
Monitoring archiver..................................................................................................................46  
Logging archiver messages......................................................................................................46  
Debugging archiver....................................................................................................................46  

**Chapter 5: Performing administration activities**.................................................................47  
Using tools available in the Start menu ..................................................................................47  
Using the AppServers Admin Console..................................................................................48  
Testing the BP Server and the BP Server Mail Listener.........................................................48  
JDBCRealm utility: populateJDBCRealm.................................................................................49  
Configuring database resource parameters...........................................................................49  
   Database parameters for JBoss............................................................................................49  
Configuring Business Process Portal.....................................................................................50  
   Configuring Business Process Portal server for Jasper report clean up.........................50  
   Disabling security warning in Business Process Portal....................................................51  
   Enabling refresh option in Portal page...............................................................................51  
   Configuring concurrent flows from multiple browser tabs or windows...........................52  

**Chapter 6: Using the Database Configuration Tool**............................................................53  
Before configuring a database...............................................................................................53  
Configuring the new database...............................................................................................54  
After configuring a database..................................................................................................58  
Reverting to the old Database...............................................................................................59  

**Chapter 7: Using the User Configuration Tool**.................................................................61  
About the User Configuration Tool.......................................................................................61  
Working with the User Configuration Tool..........................................................................61  
   Using the User Configuration Tool..................................................................................61  
   Files updated by the User Configuration Tool..................................................................63  

**Chapter 8: Using the AppPackager and AppDeployer Tools**............................................65  
About the AppPackager and AppDeployer Tools.................................................................66  
Application metadata.............................................................................................................66  
Application package details.................................................................................................68  
Setting properties..................................................................................................................68  
Working with the AppPackager and AppDeployer Tools.....................................................68
Chapter 9: Business Process Server web security.................................71
  Configuring web security.................................................................72
    Logger factory class name............................................................74
    XSS configuration for web security...............................................74
    XSRF configuration for web security..............................................75
  XSS handler implementation.........................................................76
    Sanitizing HTML text.................................................................76
    Sanitizing plain text.................................................................76
    XSS configuration.................................................................76
  XSRF handler implementation.......................................................81
    XSRF configuration.................................................................81
  Cross domain security...............................................................83

Glossary..........................................................................................85
Preface

For details, see the following topics:

- About this documentation
- User types
- Information on documentation
- Conventions used in this manual
- Product support contact information

About this documentation

This guide is part of the documentation set for Progress OpenEdge Business Process Server.

User types

Progress OpenEdge Business Process Server is a business process management system that can be used by the following types of users:

<table>
<thead>
<tr>
<th>User type</th>
<th>Responsibilities</th>
</tr>
</thead>
</table>
### Responsibilities

<table>
<thead>
<tr>
<th>User type</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Manager</strong></td>
<td>Responsible for automating business processes in a particular business domain. Business Process Portal's Management module serves as the primary interface to Business Process Server for the Manager, enabling the manager to monitor, analyze, and control business processes. Also uses the Business Process Modeler for modeling and simulation.</td>
</tr>
<tr>
<td><strong>Application Developer</strong></td>
<td>Responsible for creating customized applications for implementing business processes and developing interfaces associated with tasks. Application developers may work closely with Managers to define the requirements of an application, and determine the business processes.</td>
</tr>
</tbody>
</table>

### Information on documentation

This documentation includes information for the entire range of Progress OpenEdge Business Process Server users. In the following table, we recommend the guides that are most relevant to each type of user.

<table>
<thead>
<tr>
<th>If you are the …</th>
<th>Read the …</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application User</strong></td>
<td><em>Business Process Portal User’s Guide</em></td>
</tr>
<tr>
<td></td>
<td><em>First Steps Guide</em></td>
</tr>
<tr>
<td></td>
<td><em>Terminology Guide</em></td>
</tr>
<tr>
<td><strong>Manager</strong></td>
<td><em>Business Process Portal Manager’s Guide</em></td>
</tr>
<tr>
<td></td>
<td><em>Business Process Portal User’s Guide</em></td>
</tr>
<tr>
<td></td>
<td><em>Terminology Guide</em></td>
</tr>
</tbody>
</table>
## Conventions used in this manual

This document uses the following conventions and terminology notations.

<table>
<thead>
<tr>
<th>Convention (styles and terms)</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>Indicates titles of command buttons, check boxes, options, lists, dialog boxes, and portal page names.</td>
</tr>
<tr>
<td>file path</td>
<td>Indicates folder paths and file names.</td>
</tr>
<tr>
<td>Convention (styles and terms)</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Indicates book titles.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Represents code segments or examples.</td>
</tr>
<tr>
<td>/</td>
<td>Indicates the path in Windows environment. For UNIX environment, replace with forward slash &quot;/&quot;</td>
</tr>
<tr>
<td>OEBPS_HOME or %OEBPS_HOME%</td>
<td>Represents the installation folder of Business Process Server, C:\Progress\OpenEdge\oebpm\server.</td>
</tr>
<tr>
<td>STUDIO_HOME or %STUDIO_HOME%</td>
<td>Represents the installation folder of OpenEdge BPM components, C:\Progress\OpenEdge\oebpm\studio.</td>
</tr>
<tr>
<td>JBOSS_HOME or %JBOSS_HOME%</td>
<td>Represents the installation folder of JBOSS server, C:\Progress\OpenEdge\oebpm\jboss.</td>
</tr>
</tbody>
</table>

## Product support contact information

If the product documentation does not provide a solution to your specific issue, or if you need clarification on the issue, then contact our Product Support team. You can contact the team through the Internet, telephone, or postal mail, as per the details provided in the following table.

**Table 1: Product Support Contact Information**

<table>
<thead>
<tr>
<th>To contact by</th>
<th>Use</th>
</tr>
</thead>
</table>
| Web site        | [http://progresslink.progress.com/supportlink](http://progresslink.progress.com/supportlink)  
If you are an existing customer, then you can log in to the above site for product support. If you are a first time user, then you need to create an account first. |
| Telephone¹      | 1-781-280-4999 for US, Latin America and Canada  
1-781-280-4543 for the Product Support Fax Line |
| Postal Address¹ | Progress Software Corporation  
14 Oak Park Drive  
Bedford, MA 01730, USA. |

To enable us to quickly answer your questions, please provide the following information:

- Your name, installation site address and the license key for Business Process Server software.
- Your Business Process Server version and build number.

¹ For support telephone numbers and offices in your region, visit the support web site above. This contact information is for customer support only.
• Your operating system, application server and browser, with version and service pack details, if any.
• Your database management system and version, and information on JVM and JDBC used.
OpenEdge Business Process Server overview

Progress Software Corporation is a leading global provider of automated business process management solutions. The company’s product, Progress OpenEdge Business Process Server (henceforth referred to as Business Process Server or BP Server), is a comprehensive business process management platform, which enables companies to quickly transform their business processes into flexible and manageable Web applications, distributed over intranets, extranets, and the Internet.
Business Process Server addresses every stage in the business life cycle: define, integrate, publish, monitor, analyze, improve, and control. By adopting an end-to-end approach, Business Process Server incorporates all the key elements required to meet the ever-changing demands of e-business while ensuring e-business success. By providing integrated management tools, Business Process Server lets you monitor operations proactively, modifying automated processes dynamically based on changing external operations online. An overview of a typical automated business process management solutions is shown in Figure 1: Business Process Server overview on page 16.

**Figure 1: Business Process Server overview**

For details, see the following topics:

- Business Process Server components
- How Business Process Server works
- Business Process Server user types
Business Process Server components

Business Process Server is a suite of integrated components that enables you to easily build intranet, extranet, and Internet applications and manage your e-business. Business Process Server consists of the following components as in Figure 2: Business Process Server components on page 17:

**Figure 2: Business Process Server components**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Process Portal Home</td>
<td>The Home module of Business Process Portal is the primary interface for application users, enabling them to interact with Business Process Server applications. Users can complete entries to various tasks and applications, and link to the support infrastructure required to complete these tasks.</td>
</tr>
<tr>
<td>Business Process Portal Management</td>
<td>The Management module enables managers to query, report on, and control processes and resources, visible only to the managers.</td>
</tr>
<tr>
<td>Business Process Portal Administration</td>
<td>The Administration module enables Business Process Server Administrators to modify configuration parameters, manage user or group access control, and install or uninstall Business Process Server applications.</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Web services</td>
<td>This component allows application developers to: publish their applications as Web services, and find and convert other available Web services on the Internet into Business Process Server applications.</td>
</tr>
<tr>
<td>BPM Webflow</td>
<td>This component provides a framework for developing and implementing Web-enabled workflow applications.</td>
</tr>
<tr>
<td>Progress Developer Studio for OpenEdge</td>
<td>This is the primary application development tool for Business Process Server, which provides an Eclipse-based integrated development environment in which users can create BPM projects, processes, Web applications, and rule files.</td>
</tr>
<tr>
<td>Business Process Modeler</td>
<td>This tool is used to design templates for basic business processes, and to run simulations of processes and individual worksteps.</td>
</tr>
<tr>
<td>BP Server</td>
<td>This is a flexible, lightweight, scalable workflow process engine for intranets, extranets, and the Internet.</td>
</tr>
<tr>
<td>BPM Events</td>
<td>This open, event-driven rule engine is used to formulate and enforce policies in business applications.</td>
</tr>
</tbody>
</table>
How Business Process Server works

The following figure provides an overview of the interaction between Business Process Server components.

Figure 3: How Business Process Server works

1. Progress Developer Studio for OpenEdge and Business Process Modeler provide an integrated development environment (IDE) for Business Process Server, where you can design and publish business processes. The application developer designs a process template (with the *.spt or *.swt extension) in the IDE that reflects the business flow and other business process requirements. Business rules for the process template can be defined using the Rule Editor, a BPM Events component that is launched with Progress Developer Studio for OpenEdge.


3. Once the process template is defined, Business Process Server Administrators use the Administration module to install the business process on the BP Server. Administrators can also configure Business Process Server components, manage user or group access control, and publish Business Process Server applications as Web services. Once installed, users access applications through servlets that pass the requests over an RMI/IIOP connection to the BP Server within an EJB Container.
4. The EJB Container provides a runtime environment that executes and manages Java-based program components that run on the server side of a client/server network. Within the EJB Container are the BP Server and BPM Events server.

5. The BP Server writes events to event tables in the database. Each Business Process uses JDBC to connect to database server as well as store events in the database. Within the BP Server, BPM Process Store uses JDBC to connect to the database server process and retrieve the events deposited by the BP Server process. BPM Process Store interprets the events and populates the process tables. These populated tables are used by Business Process Portal modules.

6. Once the process template is installed as a Business Process Server application, application users can use the Home module to do the following:
   - Access applications
   - Obtain information to perform their tasks
   - Launch the application to start process instances from the BP Server

7. Once the process template is installed as a Business Process Server application, managers can use the Management module (if they have access privileges) to monitor execution of process instances and create reports. Servlets receive requests from managers and pass them onto the BP Server over an RMI/IIOP connection. Managers use the Report Builder to define management reports that retrieve information through JDBC to the database server.

8. BPM Events is a rule-based event or message processing server that loads application rules and executes them against the BP Server and/or external events or messages. This server persists data in the database for recovery and with the help of JDBC connects to the database.

9. Managed Adapters exchange information between Business Process Server applications and external applications by converting Business Process Server-specific protocol to the protocol of an external system such as a database or ERP system. When users add a Managed Adapter to a work step, they can define complex mapping between Business Process Server dataslots and adapter inputs or outputs of the external application. At runtime when the work step is executed, the Managed Adapter sets the adapter inputs and configuration, and maps the outputs to the appropriate output dataslots.

10. BPM Webflow is a run-time component that executes the presentation flows. This component provides a Model, View, Controller (MVC) paradigm for developing presentation flow-based applications and executing them in a Web container.

11. Business Process Servers Web services component allows BP Server applications to be published as Web services.


**Business Process Server user types**

There are four user types within Business Process Server, Application users, Managers, Application developers, Business Process Server administrators.

**User types**

Each Business Process Server user type is defined below:

- **Application users** — Application Users use Business Process Server applications to coordinate specific business tasks with another department within their company, with another company within their organization,
and/or with a business partner in another organization. The Home module in Business Process Portal serves as the primary interface in which Application Users run Business Process Server applications.

• **Managers** — Managers are typically experts in a particular business domain, such as quality assurance or human resources. They might need to work with managers from other groups in automating some of the business procedures that these groups share. The Management module in Business Process Portal serves as the primary Business Process Server interface for business managers to coordinate and integrate business processes, enabling them to exchange information with one other, and to share functionality over such standard communication channels as the Internet or e-mail.

• **Application developers** — Application developers are responsible for analyzing business processes and developing interfaces associated with creating tasks or processes. Application developers are often not domain experts themselves, but work closely with Managers to define business processes and determine the requirements of an application. Application developers use Progress Developer Studio for OpenEdge or Business Process Modeler to define the business process; the resulting process template file is tested, simulated, published, and run as a Business Process Server application.

• **Business Process Server administrators** — Business Process Server administrators are responsible for configuring Business Process Server components, managing user or group profiles and access control, and installing or uninstalld Business Process Server applications. The Administration module in Business Process Portal serves as the primary interface for Business Process Server Administrators to administer applications.

All Business Process Server user types can communicate by using one or more Business Process Server applications. They can also communicate with external applications.
Setting Password Security and Encryption

Business Process Server provides password security policy framework, which includes a set of interfaces. The enforcement of the password policy depends on the classes implementing the framework interfaces. The default password policy implementation is provided in Business Process Server.

For details, see the following topics:

- Password security framework
- Enabling password security in Business Process Server
- Password security default implementation

Password security framework

Business Process Server has a framework of contract, defined for Password security. The framework of contract provides the basic security options and comprehensive security policy.

Any custom implementation of this interface may store the password policy data by implementing the methods in these interfaces that enforces the policies on password security.

A short description of the contract is as follows.

Password security framework interfaces

The following are the password security framework interfaces:
• **IPasswordSecurityData** — Defines the contract for password security data. It provides methods for getting details of password policy and rules.

• **IPasswordRule** — Defines the contract for password rules. This provides methods for validating a password against defined rules. The rule is part of the password security data and the rule can be retrieved using IPasswordSecurityData.

• **IPasswordSecurityService** — Defines the contract for different services provided. It verifies whether the password is complaint with the policy and validates whether the password is compliant with the rules.

**Password security policies supported**

Business Process Server supports the following password policies.

• Change at first login
• Password Longevity
• Password Expiry Notification

You can extend the default password security. Alternatively, you can override the default password security to support a comprehensive security policy by customizing the default interfaces.
**Change at first login**

As shown in [Figure 4: Create user](#) on page 25, while creating a new user, the admin user may enable/disable **Update profile at first login** option for each user.

**Figure 4: Create user**

![Figure 4: Create user](image)

When a user logs into the portal for the first time after enabling Update profile at first login, he is redirected to the update profile page. It is mandatory for the user to change the password in the profile page as shown in [Figure 5: Change password](#) on page 25.

**Figure 5: Change password**

![Figure 5: Change password](image)
Password longevity

The password policy defines a period for which the password of all the users is valid. At the end of the longevity period, the user's password expires. During login, the user's password is checked. If the password expires, then the user is redirected to the password expired page, where the user must provide a new password.

Note: When the password expires, user cannot log in to portal until the password is changed.

Password expiry notification

In password policy you can also define the period in days before which the user should be notified about his password expiration. The password expiry notification message appears on the header section for five seconds after successful login. The notification message is displayed if the user's password is about to expire within the number of days specified for notification.

Password validation

The default password policy implementation does not provide password rules validation. If Password security is enabled and user has provided the custom implementation of the IPasswordRule interface, then the password is validated as follows:

• Whenever a new password is added for a user
• When a user's password is changed

Portal invokes the custom implemented password rules to validate the entered password. If the password is invalid as per the password rules set in the policy, then an error message is displayed.

Note: You cannot provide a password that violates any of the password rules.

The password validation feature impacts the following modules in portal.

Create user

When a new user is created, the admin user enters a password for the user. This password is validated according to the validation rules.

Change password

If you change the password in the following Portal pages, then the new password is validated at the time of submit.

• MyHome > Profile > Change Password
• Administration > User Details > Change Password

Password expired

On the password expired page, user is required to enter a new password. This password is validated at the time of submit.
Enabling password security in Business Process Server

By default the password security feature is disabled in Business Process Server. To enable this feature, you need to set the parameter `bpmportal.ps.enable` to `true` in `bpmportal.conf`.

The following parameters in `bpmportal.conf` specify the values for the classes implementing the `IPasswordSecurityData` and `IPasswordSecurityService` interfaces.

Table 3: Default handlers

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpmportal.ps.enable</td>
<td>false</td>
</tr>
<tr>
<td>bpmportal.ps.datahandler</td>
<td>com.savvion.sbm.bpmportal.security.PortalPasswordSecurityData</td>
</tr>
<tr>
<td>bpmportal.ps.servicehandler</td>
<td>com.savvion.sbm.bpmportal.security.PortalPasswordSecurityService</td>
</tr>
</tbody>
</table>

You can implement the custom handler in the following ways:

- Directly implementing the interfaces
- Extending the default handlers

Once you implement the custom handler, the property values in `bpmportal.conf` should be updated with the customized class name. By default, these properties contain the default handler names.

**Important:** Restart Portal server after you change these properties.

Password security default implementation

The following sections explains about password security default implementation.

Default handlers

The default password implementation provides the following as the default handlers.
Table 4: Default handler - Description

<table>
<thead>
<tr>
<th>Implementation class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.savvion.sbm.bpmportal.security PortalPasswordSecurityData</td>
<td>This is the default data handler that implements IPasswordSecurityData to manage the data required for password policies.</td>
</tr>
<tr>
<td>com.savvion.sbm.bpmportal.security PortalPasswordSecurityService</td>
<td>This is the default Service handler that implements IPasswordSecurityService and provides all abstract services for password security.</td>
</tr>
</tbody>
</table>

Default password policies configuration

The following are the password policies set in the default implementation.

Change at first login

If you have enabled the default password policy implementation, then the parameter, `bpmportal.ps.changeatfirstlogin` in `bpmportal.conf` specifies whether the Change At First Login policy is enabled or disabled. The default value is "true".

Password expiration

If you have enabled the default password policy implementation, then the parameter, `bpmportal.ps.longevity` in `bpmportal.conf` specifies the period (in days) for which the password is valid. The default value is 30.

Password expiry notification

If you have enabled the default password policy implementation, then the parameter, `bpmportal.ps.notifyexpiration` in `bpmportal.conf` specifies the period in days before password expiration. The default value is 3. Table 5: Password expiry notification on page 28 describes the various properties of Password Expiry Notification.

Table 5: Password expiry notification

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpmportal.ps.changeatfirstlogin</td>
<td>Flag to enable change password during the first login.</td>
<td>true</td>
</tr>
<tr>
<td>bpmportal.ps.longevity</td>
<td>Duration in days for which the password is valid. The value should be between 1 and 365. If set to -1, then password never expires.</td>
<td>30 (days)</td>
</tr>
<tr>
<td>bpmportal.ps.notifyexpiration</td>
<td>Number of days before password expiry for notification. The value should be between 1 and 30, if set to -1 then the notification is disabled.</td>
<td>3 (days)</td>
</tr>
</tbody>
</table>
Note: There is no default implementation for the password rules.
Configuring File-based Document Management System (DMS)

Business Process Server by default uses database for storing and managing documents. However, you can also configure Business Process Server to use File based DMS for storing documents. Configure Document dataslot storage before you start using Business Process Server. If you migrate from one Document dataslot storage to another, then the data cannot be migrated to the new DMS.

To configure a file based DMS with your Business Process Server, you update `<OEBPS_HOME>/conf/dms.properties` configurations as instructed below:

1. Provide the values for the following properties in `<OEBPS_HOME>/conf/dms.properties`:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dms.storage</td>
<td>Set this property to ‘fs’</td>
</tr>
<tr>
<td>dms.fs.protocol</td>
<td>This property value should be empty. For example, dms.fs.protocol=</td>
</tr>
<tr>
<td>dms.fs.root</td>
<td>Specify an existing folder on local file system where Business Process Server is installed. This folder is used for storing documents attached to Document dataslots in Business Process Server applications.</td>
</tr>
</tbody>
</table>

2. Create a folder with the name `docsroot` within the location specified by the `dms.fs.root` property in the `dms.properties` file available under the `<OEBPS_HOME>/conf` directory. The documents are stored in the specified folder on the local file system.
3. Restart all application servers and Business Process Server to use the file based DMS for Document dataslot storage.
Using the Archiver

This chapter explains what is Archiver, Archiver architecture, Archiver scheduler, Archiver client and configurations.

For details, see the following topics:

• Archiver architecture
• Data handlers
• Archiver scheduler
• Archiver client
• Archiver configuration
• Monitoring archiver
Archiver architecture

The Archiver tool periodically selects the qualified process instances that are in completed or removed state from Business Process Server database and purges them. The archiver architecture supports backing up of qualified process instances to the file system. The file system may also be a shared file system to which the completed process instance data can be stored in XML format.

**Figure 6: Archiver architecture**

Archiver handles only runtime instance data. It does not handle any template data. The archiver tool performs following operations.

- Data Extraction
- Data Storage
- Data Purging

**Data handlers**

Archiving of the data (process or event) is done with the help of Data Handlers. The Data Handlers can be classified into following types.

- Data Extractor
- Data Purger
• Data FileWriter

The Handlers are defined in the configuration file and the order of the execution of these handlers is decided by the `order` attribute.

```xml
<DataExtractor order="1" .../>
```

## Data handlers dependency

You need to follow the restrictions given below for the combinations of different types of handlers.

• Data Extractor is mandatory handler for the archiver.

• Data Extractor cannot be the only Handler. You must specify another handler, which may be a purger or file-writer, to process the extracted data.

• Data Extractor should always be the first Handler and Data Purger should always be the last Handler.

• If FileWriter Handler is defined, then the extractor should be ProcessDataExtractor.

• If Purger Handler is defined, then the extractor should be ProcessDataIdExtractor.
Data handler attributes

Table 6: Data handler attributes on page 36 lists and describes each attribute as supported by the Data Handlers.

Table 6: Data handler attributes

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Extractor</th>
<th>Purger</th>
<th>FileWriter</th>
</tr>
</thead>
<tbody>
<tr>
<td>order</td>
<td>The order of execution of the handlers.</td>
<td>The order of execution of the handlers.</td>
<td>The order of execution of the handlers.</td>
</tr>
<tr>
<td>maxRows (Any non-negative value. Default is 100.)</td>
<td>Maximum number of rows to be selected. If zero, then all the records are selected.</td>
<td>Maximum number of records purged per batch (Purging internally happens in batches and each batch is in different transaction). If zero, then all the records are purged in a single batch. The maximum value allowed is 500.</td>
<td>Not supported. This attribute is ignored.</td>
</tr>
<tr>
<td>DbStore (points to a DBStore tag defined in the XML.)</td>
<td>Database from where the data needs to be extracted.</td>
<td>Database from where the extracted data needs to be purged.</td>
<td>Not supported. This attribute is ignored.</td>
</tr>
<tr>
<td>FileStore (points to a FileStore tag defined in the XML.)</td>
<td>Not supported. This attribute is ignored.</td>
<td>Not supported. This attribute is ignored.</td>
<td>Location and the file name in which the extracted data needs to be written to.</td>
</tr>
<tr>
<td>purgeDocs</td>
<td>Not supported. This attribute is ignored.</td>
<td>If set to true, then document purging is done along with process purging. This is ignored if only events are getting purged.</td>
<td>Not supported. This attribute is ignored.</td>
</tr>
</tbody>
</table>

Data extractor handler

For any archiving operation a Data Extractor handler is mandatory. It extracts process instances and/or events from BPM Process Store tables of Business Process Server database. The data qualified for extraction is decided by the filter parameters provided by the user. The data extracted by Data Extractor is passed to any registered handlers that may be specified by the user to perform data cleansing or transformation. These handlers for data transformation or purging are optional. They accept the Data Object created by Data Extractor.

Data Extractor handler can be classified into following types:

- Handler extracting entire process instance or event data (ProcessDataExtractor)
- Handler extracting only the process instance id (ProcessDataIdExtractor)

You cannot simultaneously use both the above mentioned Data Extractors.
**ProcessDataExtractor: Extracting entire data**

The ProcessDataExtractor for extracting entire data extracts the entire data for all the qualified process instances or events. It is recommended to use the extractor in combination with ProcessDataFileWriter, which is configured to copy data to any target file in XML format.

The extracted data is temporarily stored in value Objects.

- **ProcessInstance value object**, which stores the process instance graph of a single process instance, is composed of the following value Objects:
  - WorkStepInstance
  - WorkItem
  - CStep
  - CWorkItem
  - CStepDS
  - WIResourceUsage
  - SubProcessInfo
  - Map containing data fields.
  - ProcessNotes
  - RuntimeAlerts

- **Event value object**, which stores the contents of a single event. In some cases, this value object may become part of composition of ProcessInstance value object.

The ProcessInstance value object is created from the following database tables.

- PROCESSINSTANCE
- SUBPROCESSINFO
- WORKSTEP
- WORKITEM
- CWORKITEM
- CSTEP
- CSTEP_DS
- PROCESS_NOTES
- WI_RESOURCE_USAGE
- RUNTIME_ALERT
- DataFields or Dataslots (instance values only, no global values)

The Event value object is created by selecting data from BIZEVENT table.

- BIZEVENT
**ProcessDataIdExtractor: Extracting only Id values**

The ProcessDataIdExtractor for extracting only Id values extracts only the Ids of all the qualified process instances. It is recommended to use the extractor in combination with:

- ProcessDataPurger, which is configured to purge the extracted data.

This extractor extracts only the Ids of Process Instances to avoid occupying a lot of memory by loading the entire graph of all qualified process instances. During process data extraction, along with process instance Ids, instance data field (dataslot) table name is also extracted and passed to Data Purg to enable purging of data fields.

**Data purging handler**

The Data Purging Handler deletes the data pertaining to all the qualified process instances or events extracted by the Data Extractor handler. Data belonging to all Process Instances/Events is not deleted in one single transaction. The data is divided into batches of fixed size, and each batch deletion happens as a separate transaction.

All data of a qualified process instance that is in a completed or removed state is deleted from BPM Process Store. The purger deletes all data pertaining to a process instance including worksteps, workitems and the instance data fields. It also deletes the process alerts that are in deleted state.

**ProcessDataPurger**

It is recommended to use the ProcessDataPurger handler in combination with ProcessDataIdExtractor, which extracts only ids of qualified process instances. The purging operation is carried in three different ways depending upon the data extracted by the Data Extractor handler.

- Purging process instance data with events
- Purging process instance data only
- Purging events only

When event purging is enabled, it is checked if the events are already processed by other components like BPM Process Store, BPM Events and EventPublisher (whichever components are active and are configured in Archiver configuration file).

Default XML configuration files used for event data purging and process instance data purging are provided below for reference. These are the default configuration files used by the archiving schedulers for purging.

**Purging of event data**

In this example, the process instance Ids are extracted and events for those process instances are purged.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ArchiverConfig>
<!-- Please make sure correct values are provided for the below options. If BP servers (EventPublisher or BPM Events) are running, then change the value for the attribute 'running' accordingly. -->
<BMServers>
  <EventPublisher running="false"/>
  <BPM Events running="false"/>
</BMServers>
<!-- The mentioned handlers are executed in the specified order -->
```
The above xml is used for extracting and purging for events. Please note the following points about the xml contents.

- The extraction is done by ProcessDataIdExtractor, which extracts maximum 10000 records of Process Instance Ids, defined by the maxRows attribute.
- The extracted data is purged in batches of size 50 (that is, events belonging to 50 Process Instances). This is set again by maxRows attribute.
- Both the handlers use the DBStore BMDatabaseStore to connect to the database. This is a default datasource created for connecting to Business Process Server database. The DBStore BMDatabaseStore uses direct connection to connect to the database.

**Purging of process data**

In this example, the process instance Ids are extracted and data for those process instances are purged.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ArchiverConfig>
  <!-- Please make sure correct values are provided for the below options. If BP servers (EventPublisher or BPM Events) are running, then change the value for the attribute 'running' accordingly. -->
  <BMServers>
    <EventPublisher running="false"/>
    <BPM Events running="false"/>
  </BMServers>
  <!-- The mentioned handlers are in the specified order -->
  <DataHandlers>
    <DataExtractor order="1" maxRows="10000" dbStore="BMDatabaseStore">
      com.savvionbm.archiver.handlers.ProcessDataIdExtractor
    </DataExtractor>
    <DataPurger order="2" maxRows="50" dbStore="BMDatabaseStore">
      com.savvionbm.archiver.handlers.ProcessDataPurger
    </DataPurger>
  </DataHandlers>
  <!-- Direct DBStore used to point to BP Server database -->
  <DBStore name="BMDatabaseStore" type="Direct" useBMStore="true"/>
</ArchiverConfig>
```

The above xml is used for extracting and purging for events. Please note the following points about the xml contents.

- The extraction is done by ProcessDataIdExtractor, which extracts maximum 10000 records.
- The extracted data is purged in batches of size 10.
- Both the handlers use the DBStore BMDatabaseStore to connect to the database. This is a default datasource created for connecting to Business Process Server database. The DBStore BMDatabaseStore uses direct connection to connect to the database.
Data FileWriter handler

Data FileWriter should be configured when the qualified instances from Business Process Server database should be written to a file in XML format. The default writer writes data extracted by the Data Extractor to a single archive file. XStream is used to covert the data into XML format. The handler ProcessDataFileWriter performs the file writing operation.

ProcessDataFileWriter

The ProcessDataFileWriter handler always works in combination with ProcessDataExtractor, which extracts the entire process instance or event data. All data of a single process instance including worksteps, workitems and application data is written within a single XML tag. This may be useful for transformation to HTML format and viewing it from a browser. But it may not be useful for data warehouse applications. ETL tools require data of a single database table to be stored together for easy data loading. All the data of the qualified process instances or events are stored in ProcessData_<jobid>.xml file. The jobid is unique for each archiver run.

Sample Process XML generated by Data FileWriter

A snapshot of process XML structure is given below.
<list id="1">
<com.savvion.bm.archiver.models.ProcessInstance id="2">
<workSteps id="5">
<com.savvion.bm.archiver.models.WorkStepInstance id="6">
<name>Evaluate Resume</name>
<workItems id="9">
<com.savvion.bm.archiver.models.WorkItem/>
</workItems>
</com.savvion.bm.archiver.models.WorkStepInstance>
</workSteps>
<events id="21">
<com.savvion.bm.archiver.models.Event id="22"/>
</events>
</com.savvion.bm.archiver.models.ProcessInstance>
</list>

Sample Event XML generated by Data FileWriter

A snapshot of event XML structure is given below:
<list id="1">
<com.savvion.bm.archiver.models.Event id="2">
<id>1</id>
<eventType>P_CREATED</eventType>
<eventCreator>BP_Server</eventCreator>
<createdDate>1219992776519</createdDate>
<instanceId>0</instanceId>
<templateId>28</templateId>
<workStepId>0</workStepId>
<context class="hashtable" id="3">
<entry>
<string>PROCESSTEMPLATENAME</string>
<string>Hiring</string>
</entry>
</context>
</com.savvion.bm.archiver.models.Event>
</list>
Archiver scheduler

Process and event data archiving can be scheduled using Quartz scheduler. This scheduler runs within Business Process Portal Server.

The archiver scheduler has two predefined jobs; purging event data and purging process data. Once a process purge job is activated, the scheduler makes sure that no other process purge job gets fired before the completion of the current job. Same is the case with event purge job also.

Quartz scheduler configuration

The Quartz scheduler uses the parameter values explained in the following table.

Table 7: Quartz scheduler parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceId</td>
<td>AUTO</td>
</tr>
<tr>
<td>ThreadPoolClass</td>
<td>SimpleThreadPool</td>
</tr>
<tr>
<td>Trigger</td>
<td>CronTrigger</td>
</tr>
<tr>
<td>Misfire</td>
<td>CronTrigger.MISFIRE_INSTRUCTION_DO NOTHING</td>
</tr>
<tr>
<td>JobStore</td>
<td>JobStoreTX</td>
</tr>
<tr>
<td>Jobs</td>
<td>Stateful Jobs //concurrency is not required</td>
</tr>
<tr>
<td>Job Persistence</td>
<td>Durability=true</td>
</tr>
<tr>
<td>Locking</td>
<td>UpdateLockRowSemaphore</td>
</tr>
<tr>
<td>Locking Query</td>
<td>UPDATE (0)LOCKS SET LOCK_NAME = LOCK_NAME WHERE LOCK_NAME = ?</td>
</tr>
</tbody>
</table>

Archive scheduler configuration

The Archiver scheduler is configured using the property file archivescheduler.properties. The parameter values explained in the following table are used for the archive scheduler.

Table 8: Archiver scheduler parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>archive.scheduler.process.delete</td>
<td>false</td>
<td>This property defines whether the Scheduler to purge process data needs to be enabled.</td>
</tr>
</tbody>
</table>
### Persistence of archiver jobs

You can use the command line utility `schedulearchivejobs.cmd|sh` from the OEBPS_HOME\bin directory to persist scheduler jobs and associated triggers in database. This utility reads jobs and trigger information from the configuration file `archivescheduler.properties` to create the jobs and triggers, and persists them in the Business Process Server database.

It is recommended that you should modify the required properties for jobs and trigger in `archiverscheduler.properties` and invoke this tool to persist the same. For more information about the parameters, refer to the Archive scheduler configuration on page 41.

To the persist scheduler properties you need to execute the `schedulerarchivejobs.cmd|sh` command before starting the portal server.

### Scheduler startup and shutdown

The following properties in `archivescheduler.properties` should be set to `true` to enable the scheduler for process and events.

- `archive.scheduler.event.delete=true`
- `archive.scheduler.process.delete=true`

After updating these properties, run the `schedulerarchivejobs.cmd|sh` utility.

The archive scheduler is initialized during the Portal server startup. The ‘init’ method of a Servlet initializes the scheduler. During initialization, the scheduler reads `archivescheduler.properties` configuration file to check whether process and/or event scheduling is enabled. If any of them are enabled, then the scheduler picks up job and trigger information from Business Process Server database and schedules those jobs. When portal server is shut down, the Scheduler is also shut down.
Archiver client

You can invoke archiver as a command line utility using one of the following command script files from OEBPS_HOME\bin directory. In all these scripts, the option `--config` is mandatory, which is used to specify full path of the archiver XML configuration file.

```
archiveevents.cmd/sh
archiveprocess.cmd/sh
archiveprocesswithevents.cmd/sh
```

Note that with all command line options, the archiving or purging of data happens immediately on invoking the command line scripts. No quartz jobs are scheduled for any command line requests.

The command line options for these scripts are described below.

```
[-config <arg>]
```

Use this option to specify archiver XML configuration file with its complete path. This is a mandatory option.

For example, `--config c:/oebps/conf/archiverprocesspurge.xml`

```
[-delete -ptname ]
```

For `-delete` option, it is mandatory to provide a Process Template name with `-ptname` option. If this option is specified, then the extracted data is deleted from database.

Also note that with `-delete` option, you cannot use the `-idrange` and `-piidlist` options. Also, with `-delete`, for `-pistatus` option valid values are `PI_COMPLETED` and `PI_REMOVED`.

```
[-databefore <int>d]
```

Use this option to specify the number of days. The data older than these many number of days is archived (and possibly deleted) and `int` is an integer number.

For example, `--databefore 90d` //to archive data older than 90 days.

```
[-idrange <arg>]
```

Use this option to specify the range of either process instance ids or event ids for archiving. This option specifies event-ids for `archiveevents.cmd/sh` utility and process-ids for `archiveprocesswithevents.cmd/sh` utility. It cannot be used along with `-piidlist` option.

For example, `--idrange 200-250` //archives pis or events between Ids 200 and 250, both numbers inclusive.

```
[-piidlist <arg>]
```

Use this option to specify comma-separated values of Process Instance Ids for archiving. This option cannot be used along with `-idrange` option.

For example, `--piidlist "130,145,189"` //archives data of Process Instance Ids:130, 145 and 189.

```
[-pistatus <arg>]
```

Use this option to specify comma-separated values of process instance status. Valid status values are `PI_ACTIVATED`, `PI_SUSPENDED`, `PI_COMPLETED`, and `PI_REMOVED`. Note that if this option is not provided, then the default value is `PI_COMPLETED` or `PI_REMOVED` for -processonly and -processwithevents. If the option is -eventsonly then no value is considered by default.
But if -delete option is specified by the user, then for options -eventsonly, -processonly and -processwithevents the values supported are PI_COMPLETED and/or PI_REMOVED and if it is not provided then the default value considered is PI_COMPLETED and PI_REMOVED.

For example, -pistatus "PI_COMPLETED,PI_ACTIVATED" // archives complete and activated pis.

[-ptname <arg>]

Use this option to specify Process Template Name to archive data related only one particular process template.

For example, -ptname "Hiring" //archives all instance data of ‘Hiring’ process template.

**Note:** To archive large volume of data, and store it to the file system, you may need to adjust the maximum Heap memory in the command line scripts.

[-allpt <arg>]

Use this option to archive data for all process templates.

**Note:** The -allpt works with only -delete option. Apart from all mandatory options, if you specify either -allpt or -ptname with -delete, then the events get deleted as defined.

---

**Archiver configuration**

Archiver can be configured to invoke handlers in a specified order. A sample archiver configuration file is shown below.

**Configuration**

<ArchiverConfig>
  <BMServers>
    <EventPublisher running="false"/>
    <BPM Events running="false"/>
  </BMServers>
  <DataHandlers>
    <DataExtractor order="1" maxRows="100" dbStore="BMDatabaseStore"/>
    com.savvion.bm.archiver.handlers.ProcessDataExtractor
  </DataExtractor>
    <DataFileWriter order="4" fileStore="FileStore"/>
    com.savvion.bm.archiver.handlers.ProcessDataFileWriter
  </DataFileWriter>
    <DataPurger order="5" dbStore="BMDatabaseStore"/>
    com.savvion.bm.archiver.handlers.ProcessDataPurger
  </DataPurger>
  </DataHandlers>

<!-- The FileStore used by Writer to store data in XML format -->
<FileStore name="FileStore">
  <Property name="fileNamePrefix" value="ProcessData"/>
  <Property name="location" value="D:\swinstall\oebps\bm71pro10g\archiver\archivedata"/>
</FileStore>

<!-- To be used when executed within the container -->
<DataSourceStore type="DataSource">
  <Property name="datasource" value="jdbc/SBMCommonDB"/>
</DataSourceStore>

<!-- Direct DBStore used to point to BP Server database -->
<DBStore name="DataSourceStore" type="DataSource">
  <Property name="datasource" value="jdbc/SBMCommonDB"/>
</DBStore>
</ArchiverConfig>
You can find the following sample archiver configuration files in the OEBPS_HOME\archiver\samples directory.

- archiverfilewrite.xml
- archiverpurge.xml

**BM Servers tag**

This tag in archiver XML configuration file is relevant for purging events. EventPublisher, BPM Events and BPM Process Store are the different BM Servers, which process events. They process all the events that are fired in Business Process Server. An event can be purged only when all these BM Servers have completed processing of that event. These components, except BPM Process Store, are optional, and may or may not get used in a Business Process Server system. Thus, checking whether they have already processed an event can be avoided by specifying value for running attribute as false.

**DataHandlers tag**

The <DataHandlers> tag specifies the list of handlers to be invoked as a part of archiving process. The handlers are invoked in the specified order. But there are some restrictions in the way that one can specify the order and the handlers. The DataExtractor handler should not be invoked after Data Writer handlers. For more information, refer to Data handlers dependency on page 35.

**FileStore and DBStore tags**

The <FileStore> tag is used to specify file properties of storing data into XML format into these files. DataFileWriter handler can use the FileStore name.

A <DBStore> tag is used to specify a database account. This tag has following attributes - name, type and useBMStore. The valid values for these attributes are as follows.

- **name** — any valid text string
- **type** — Direct, DataSource
- **useBMStore** — true, false

Note that to point to Business Process Server database account, it is required to set the useBMStore attribute to true. When it is set to true, Business Process Server database information is extracted from the oebpsdb.properties configuration file. When the useBMStore attribute is set to false and type attribute is Direct, then you need to provide the URL, driver, user name and password property values. When the type attribute is set to DataSource, then you need to provide the property datasource with a valid datasource value. One handler can refer to only one store.
Monitoring archiver

The details about each archiver run including archive run job Id, archiving options, configuration and other relevant information are written to the ArchiveRun_<jobid>.xml file, stored in the OEBPS_HOME\archiver\archivedata directory. The jobid is unique for each archiver run.

Logging archiver messages

Archive Scheduler is initialized when Portal server starts. This can be verified by looking for the following entry in bpmportal.log file.

...(PW14051):Archive Scheduler service started successfully....

The file archiver.log provides the progress and monitoring details of the archive jobs and command line executions.

Debugging archiver

You can enable debug information, by changing the property log4j.category.Archiver in the oebpslog.conf file from info,Archiver.File to debug,Archiver.File. You need to restart the portal server for this change to be effective in schedulers. However, it is effective immediately for all command line scripts.
Performing administration activities

This chapter describes the administration activities you may need to perform after completing the Business Process Server installation.

For details, see the following topics:

• Using tools available in the Start menu
• Using the AppServers Admin Console
• Testing the BP Server and the BP Server Mail Listener
• JDBCRealm utility: populateJDBCRealm
• Configuring database resource parameters
• Configuring Business Process Portal

Using tools available in the Start menu

Business Process Server provides several tools in the Start menu that can help you to configure your installation or provide supplemental tools for specific features. You can navigate to the tools using Programs > Progress > OpenEdge > BP Server > Tools. These tools include:

• **DB Configuration Tool** is used to configure a new database with Business Process Server. For more information, see Using the Database Configuration Tool on page 53.

• **User Configuration Tool** is used to modify Application Server, LDAP, and e-mail user details. For more information, see Using the User Configuration Tool on page 61.
Using the AppServers Admin Console

Using the AppServers Admin Consoles users with the required permissions can install, configure, monitor, manage, and deploy one or more application servers.

Perform the following:

1. To access the AppServer Console for JBoss, you can use the following shortcut: Programs > Progress > OpenEdge > BP Server > Admin Consoles.
   Then select either Portal Console or Ejb Console to open the Server Management Console GUI for the application server.

2. Enter Username and Password. The Username and Password varies according to your application server. For example:
   - For JBossEnterprise 5.1.1 or JBoss Community Edition 5.1, log into the Admin Console of either Portal Console or Ejb Console with the user name "guest" and the password "guest".

3. Use the functions in the Management Console to:
   - Create views on multiple servers.
   - Perform remote, unified management of application servers and applications published on them.
   - Seamlessly publish applications on servers.
   - Remotely publish applications across firewalls.
   - Create special alert messages that can be received on any device.
   - Maintain a high level of security that supports JAAS framework.

Testing the BP Server and the BP Server Mail Listener

When a human activity workstep is configured as "Send mail and complete its reply" then BP Server sends an e-mail to the user(s) when that workstep is activated. When the user replies, then the BP Server Mail Listerner reads that e-mail and on behalf of user completes the task.

Depending on your individual setup, the Incoming Mail Server and the Outgoing Mail Server may reside on the same computer, or may reside on different computers. The configuration settings for these servers are contained in oebpsemail.properties file in OEBPS_HOME\conf\ folder.

To enable BP Server mail listener, you have to set the bpserver.email.reader.start=True in BP Server.conf file in OEBPS_HOME\conf\ folder. The default value for this parameter is true.

---

Important: You have set this parameter before starting the BP Server.

Even if bpserver.email.reader.start parameter is set as False in bpserver.conf file, then you can start the BP Server E-mail Listerner using "StartEmailListerner" command in BP Server admin utility. In the same way, irrespective of the bpserver.email.reader.start parameter value, you can stop the BP Server E-mail Listerner using "StopEmailListerner" command in BP Server admin utility.

To check if the BP Server server is running without starting a Business Process Server session perform the following:
1. Navigate to the OEBPS_HOME\bin directory and open startBP ServerAdmin.sh/cmd.
2. Run getServerState command in console to return server state.

**JDBCRealm utility: populateJDBCRealm**

This utility imports user information from previous realms into the BP Server DB-Based Realm (JDBCRealm). To run this utility, enter the following:

```
populateJDBC [-encrypt] <realmName> -force
```

<realmName> represents the previously used realm that the user wants to import data from.

Parameters for other realms require proper setup in the umacl.conf file before running this utility. For example, if you want to run the utility and take information from NISRealm, then enter as shown below:

```
populateJDBC [-encrypt] NISRealm -force
```

Then set the NISServer and NISDomain parameters in the umacl.conf file to the correct values.

**Configuring database resource parameters**

This section presents the default parameters of database resources for the application servers supported in BP Server. If you have changed database resource parameters and want to tune these parameters back to the default values, then refer to the default settings provided in the following sections.

DataSource is used to get connections for performing database operations. In BP Server, DataSource can be of two types - XA and non-XA. The DataSource files contain the database connection details and other configuration parameters. These configurations are different for different application servers. For some application servers, these changes can be done manually or can be done using some GUI provided by application server.

**Database parameters for JBoss**

To tune the database parameters for JBoss Enterprise Application Platform 5.1.1 Ejb Server and Portal Server, go to JBOSS_HOME\server\<servername>\deploy\bmjdbc directory. In this directory there are different files for different datasources.

**Datasource list for JBoss EJB server**

<table>
<thead>
<tr>
<th>DataSource Name</th>
<th>DataSource type</th>
<th>Initial/min</th>
<th>max</th>
<th>Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>jdbc/JPEServerDB</td>
<td>XA</td>
<td>10</td>
<td>40</td>
<td>150</td>
</tr>
<tr>
<td>jdbc/JPEServerDB</td>
<td>XA</td>
<td>2</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>jdbc/BPM Process StoreDB</td>
<td>NonXA</td>
<td>2</td>
<td>5</td>
<td>150</td>
</tr>
<tr>
<td>jdbc/JMSAdapterDBXA</td>
<td>XA</td>
<td>2</td>
<td>15</td>
<td>150</td>
</tr>
</tbody>
</table>
### Datasource list for JBoss Portal Server

#### Table 10: Datasource list for portal server

<table>
<thead>
<tr>
<th>DataSource Name</th>
<th>DataSource type</th>
<th>Initial/min</th>
<th>max</th>
<th>Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>jdbc/SBMPortalsDB</td>
<td>NonXA</td>
<td>5</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>jdbc/SBMPortalDB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdbc/XMLDocRepositoryDB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdbc/BizRulesDB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdbc/SMCommonDB</td>
<td>NonXA</td>
<td>3</td>
<td>20</td>
<td>50</td>
</tr>
</tbody>
</table>

### Configuring Business Process Portal

You can configure Business Process Portal for the following:

- Jasper report clean up
- Disabling security warning in Business Process Portal
- Enabling refresh option in portal page
- Configuring Business Process Portal for editable dataslot
- Current flows from multiple browser tabs or windows

### Configuring Business Process Portal server for Jasper report clean up

When you execute a Jasper report, it generates temporary artifacts in the Business Process Server's web application directory. The size of these artifacts depends on the volume of data queried by the Jasper report. Over a period of time, as the number of executed Jasper reports increases, more artifacts are accumulated in the web application directory, which occupies a considerable amount of disk space.

To overcome this issue, you can configure your Business Process Portal server to schedule a job to periodically delete the old artifacts. You can configure the server by using the following parameters from the bpmportal.conf file in the $OEBPS_HOME\conf directory.
Table 11: Parameters for Jasper report clean up

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Allowed values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpmportal.jasper.schedule.cleanup.start</td>
<td>on, off</td>
<td>If you set this parameter value as 'on', then Jasper report clean up scheduler is scheduled to run when the Portal server is started.</td>
</tr>
<tr>
<td>bpmportal.jasper.schedule.cleanup.interval</td>
<td>Must be a valid number in seconds.</td>
<td>This parameter specifies the interval between two successive Jasper report clean up jobs.</td>
</tr>
<tr>
<td>bpmportal.jasper.schedule.cleanup.startdelay</td>
<td>Must be a valid number in seconds.</td>
<td>This parameter specifies the delay in seconds to start the Jasper report clean up job after Business Process Portal server is started.</td>
</tr>
<tr>
<td>bpmportal.jasper.schedule.cleanup.cutoff</td>
<td>Must be a valid number in seconds.</td>
<td>This parameter specifies the criteria in terms of seconds for deleting the files based on their age. For example, if it is set to 3600 seconds, then the temporary files older than one hour are deleted. The default value is 86400 seconds that is, one day.</td>
</tr>
<tr>
<td>bpmportal.jasper.schedule.cleanup.timeout</td>
<td>Must be a valid number in seconds.</td>
<td>This parameter specifies how long the Jasper report clean up job can run.</td>
</tr>
<tr>
<td>bpmportal.jasper.schedule.cleanup.maxfilesize</td>
<td>Must be a valid number in MBs.</td>
<td>This parameter specifies the maximum file size allowed for deletion. The Jasper report clean up job does not delete the files with size greater than the specified value.</td>
</tr>
</tbody>
</table>

Disabling security warning in Business Process Portal

When portal server is configured with Apache-SSL proxy server, security warning is displayed on accessing dashboard, psv, and heatmap.

In order to disable security warning, set the following parameters in `bpmportal.conf` file in `OEBPS_HOME/conf` folder to blank as listed in the following table.

Table 12: Disabling security warning

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Set to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpmportal.codebase.javaulr</td>
<td><a href="http://java.sun.com/update/1.5.0/jinstall-1_5-windows-i586.cab#Version=1,5,0,0">http://java.sun.com/update/1.5.0/jinstall-1_5-windows-i586.cab#Version=1,5,0,0</a></td>
<td>(set to blank)</td>
</tr>
<tr>
<td>bpmportal.codebase.flashurl</td>
<td><a href="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0">http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0</a></td>
<td>(set to blank)</td>
</tr>
</tbody>
</table>

Enabling refresh option in Portal page

To enable refresh option in Portal page, in `bpmportal.conf` file located in `OEBPS_HOME/conf` folder enable `refresh.On.Save` parameter.
Configuring concurrent flows from multiple browser tabs or windows

Business Process Portal supports processing flows from multiple browser windows or multiple tabs in the same browser window. This enables users to open each task in a separate browser tab or window, and work on it. It provides flexibility to users to execute multiple flows simultaneously. A process can also be designed to open multiple tasks in iframes. This feature is disabled by default. If you want to enable it, then set the value of the bpmworkflow.multisession.filter.enable parameter in the `<OEBPS_HOME>/conf/bpmebflow.conf` file to TRUE. This value is case-sensitive.
Using the Database Configuration Tool

While working with the existing installation of Business Process Server, you can choose to use another database and point Business Process Server setup to it. You can configure the new database with Business Process Server using the Database Configuration Tool. This is a special Business Process Server Tool and can be accessed from the Start menu. The Database Configuration Tool changes the database-related properties in all Business Process Server *.conf and *.properties files, and uses appserver MBeans to delete and recreate the appserver data sources with the new properties.

For details, see the following topics:

• Before configuring a database
• Configuring the new database
• After configuring a database
• Reverting to the old Database

Before configuring a database

Before you configure the new database:

1. Ensure that the new database is correctly installed and a valid user is created.
2. Stop the servers in the correct sequence, according to the application server you are using.
3. Start the servers in the correct sequence, according to the application server you are using.
Configuring the new database

To open the Database Configuration Tool:

1. Select Programs > Progress > OpenEdge > BP Server > Tools > DB Configuration Tool.

   Alternatively, you can use a command prompt to start the DB Configuration Tool in Windows and UNIX.
   a) In the command prompt, go to OEBPS_HOME\bin.
   b) Enter the following command:

      • on Windows
        DBConfigTool.cmd
      • on UNIX
        DBConfigTool.sh
A **Prerequisite** window is displayed as shown in the following figure. The information displayed in this window contains instructions to be followed before and after using the Tool and is specific to the application server with which Business Process Server is installed.

**Figure 7: Database Configuration Tool, instructions window**

2. Click **Next**, displaying the Current Database Configuration window, as shown below.

**Figure 8: Database Configuration Tool, summary of Current Database Configuration**
3. Click **Next**. The New Database Properties window is displayed as shown in the following figure.

**Figure 9: Database Configuration Tool, New Database Properties**

4. Select the DBMS you want to configure from the **Database** drop-down list.

The Database Configuration Tool displays appropriate fields for the properties of the selected database. The **New Database Properties** window is shown in the **Figure 9: Database Configuration Tool, New Database Properties** on page 56 as an example. The window for other database management systems are slightly different.

If you need your database server in multi-byte mode, then you can set the Multibyte Schema option to **True**. Enter the database properties to construct the database URL.

**Note:** Check the **Use SSL** check box to connect using SSL security.

5. Click **Next**.
The **New Database User Credentials** window is displayed as shown in the following figure.

**Figure 10: Database Configuration Tool, New Database User Credentials**

a) Enter a valid User Name and Password.

b) Click **Test Connection** to validate the database details entered. On successful validation, a confirmation message is displayed. If you enter invalid details, then an error message is displayed.

**Note:** If you are using JBoss application server, the New Database Configuration Details page appears. Click **Finish**.

6. Click **Next**.
The **Machine and Server Details** window for the selected database is displayed, as shown in the following figure.

**Figure 11: Database Configuration Tool, Business Process Server Host Details**

![Database Configuration Tool](image)

7. Enter the details. Ensure that your port specification is valid, and does not conflict with the other servers.

8. Click **Finish**. A prompt appears, asking you to ensure to run the OEBPS setup tool and then start the Portal servers. Click **OK**.

**Figure 12: Server prompt**

![Server prompt](image)

**After configuring a database**

After you have successfully configured the new database using DB configuration tool and tested the data sources:

1. Remove the temporary folders from servers, for example  
   JBOSS_HOME/jboss-as/server.ejbServer/tmp,  
   JBOSS_HOME/jboss-as/server.ejbServer/log,  
   JBOSS_HOME/jboss-as/server.ejbServer/work directories.

2. Stop the servers with stopServer command. Do not use CTRL + C.

3. Run `setupOEBPS.cmd|sh`.

4. Start the servers again in the required sequence.

Business Process Server is now ready with the new database.
Reverting to the old Database

The Database Configuration Tool retains the original configuration files and also the subsequent ones, even after multiple runs of the tool, since every time the file is backed up with a unique name by appending the suffix "_<timestamp>" to the file name, where the <timestamp> is the current time in long number format. In case of JBoss application server, the Database Configuration Tool changes the datasource configuration file name as follows:

Original Filename — <datasource_file>-ds.xml
Backed up Filename — <datasource_file>-<timestamp>.xml.orig

In case you need to revert to your original setting, follow these instructions.

1. Stop all the servers.
2. Rename the following files to their original names:
   - OEBPS_HOME/conf/oebpsdb.properties_<timestamp> to oebpsdb.properties
   - OEBPS_HOME/conf/oebps.conf_<timestamp>, if exists, to oebps.conf
   - OEBPS_HOME/adapters/db/conf/db.properties_<timestamp> to db.properties

3. For JBoss and Embedded JBoss, rename the <datasource_file>-<timestamp>.xml.orig files to <datasource_file>-ds.xml in the following directories:
   - JBOSS_HOME/server.ejbServer/deploy/bmjdbc
   - JBOSS_HOME/server/portalServer/deploy/bmjdbc

4. In the case of Enterprise JBoss 5.1.1 application server:
   a) Delete the bm-<current_database_name>-persistence-service.xml file from the JBOSS_HOME/server.ejbServer/deploy/bmjms folder.
   b) Copy the target database file from the OEBPS_HOME/jboss/jms/<TARGET_DATABASE> folder to the JBOSS_HOME/server.ejbServer/deploy/bmjms directory, wherein TARGET_DATABASE is the original database.
Using the User Configuration Tool

This chapter explains what a User Configuration Tool is, how to use it, and what it does.

For details, see the following topics:

• About the User Configuration Tool
• Working with the User Configuration Tool

About the User Configuration Tool

Business Process Server, and Process Asset Manager (PAM) provides the User Configuration Tool, enabling the Business Process Server Administrator/users to update user details in various configuration files.

You can change the user name and password specified during Business Process Server installation by using the User Configuration Tool.

Working with the User Configuration Tool

The following sections explain how to work with user configuration tool.

Using the User Configuration Tool

To update the user details using the User Configuration Tool:
1. On Windows, use one of the following shortcuts to start the User Configuration Tool depending on the product from which you are starting the tool.

   - For Business Process Server, select Programs > Progress > OpenEdge > BP Server > Tools > User Configuration Tool.

Alternatively, you can use a command prompt to start the User Configuration Tool in Windows and UNIX.

   a) In the command prompt, go to the appropriate directory depending on the product from which you are starting the tool.

   - For Business Process Server, go to OEBPS_HOME\bin

   b) Enter the following command:

      - on Windows
        userconfig.cmd
      - on UNIX
        userconfig.sh

The User Configuration Tool is displayed as shown in the Figure 13: User Configuration Tool on page 62.

**Figure 13: User Configuration Tool**

2. Select the User Type from the drop-down list.

**Table 13: User types**

<table>
<thead>
<tr>
<th>User type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>Administrative User details to connect to Application Server.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Ensure that Administrative user for the respective Application Server is added before modifying the Application server credential through User Configuration Tool. For more information on adding users refer to respective Application Server documentation.</td>
</tr>
<tr>
<td>Incoming Email Server</td>
<td>User details to connect to the mailbox of bpsserver.email.id on IMAP Server.</td>
</tr>
</tbody>
</table>
All these user types are available in the User Configuration Tool started from Business Process Server. However, for the User Configuration Tool started from Process Asset Manager, LDAP and Business Process Server Admin Credentials are the only user types available. For user type Application Server, after using this tool, you also need to perform application server specific steps to update the application server specific files. For more information, refer to your application server documentation.

When no user name and password is specified while running the setupOEBPS utility or starting the BP Server, the user name and password specified for the Business Process Server Admin Credentials is used by default.

For more information, see Files updated by the User Configuration Tool on page 63.

3. Specify the user name and password in the appropriate text boxes. You need to re-enter the password in the Confirm Password text box.

   **Note:** The user name must not contain any special characters except ‘.’ and ‘@’.

4. To save the information, click **Submit**. To reset the form and remove the information you just entered, click **Reset**.

### Files updated by the User Configuration Tool

When you specify user details using the User Configuration Tool, it stores these details in various files. The following table explains the file names and the parameters updated by this tool.

**Table 14: Files and parameters updated by User Types**

<table>
<thead>
<tr>
<th>User type</th>
<th>Configuration file(s)</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Application Server         | oebpsjndi.properties  | • oebps.<appserver_name>.principal
• oebps.<appserver_name>.credentials
• oebps.<appserver_name>.jms.principal
• oebps.<appserver_name>.jms.credentials
• oebps.qs.principal
• oebps.qs.credentials
| boot.properties            |                       | • user
• password                |
### User type | Configuration file(s) | Parameters
--- | --- | ---
Incoming Email Server | oebpsemail.properties | - bpserver.email.incoming.server.user  
- bpserver.email.incoming.server.password

Outgoing Email Server | oebpsemail.properties | - oebps.email.outgoing.server.user  
- oebps.email.outgoing.server.password

LDAP | umacl.conf | - ldap.user.name  
- ldap.user.password

Business Process Server Admin Credentials | bmboot.properties | - user  
- password

The User Configuration Tool retains the original configuration files and also the subsequent ones, even after multiple runs of the this tool, since every time the file is backed up with a unique name by appending the suffix "_<timestamp>" to the file name, where the <timestamp> is the current time in long number format.
Using the AppPackager and AppDeployer Tools

After you develop Business Process Server applications, you need to package these applications into a single ZIP or JAR file, before sending them to your Business Process Server administrator. The administrator needs to unpack the ZIP or JAR file in order to publish the applications on the production systems.

This method of packaging and publishing Business Process Server applications manually can be error-prone because of the following reasons:

• While packaging the applications, you need to ensure that all the dependant JAR files, third-party libraries, adapter classes, and generated files are included.

• During publishing, the Business Process Server administrator needs to ensure that all BP Server and Web subprocesses are packaged along with the parent applications.

• The administrator must also ensure that all the application folders are copied.

In order to prevent these errors, Business Process Server has developed a command line tool that automates the process of packaging and publishing applications. In this chapter, you will learn about this command line tool and how you can use this tool to package and publish Business Process Server applications.

**Note:** After developing the application using Progress Developer Studio for OpenEdge, we recommend you to build the project (in Progress Developer Studio for OpenEdge, click **Project > Build Project**) before trying to package it.

For details, see the following topics:

• About the AppPackager and AppDeployer Tools
About the AppPackager and AppDeployer Tools

The command line tool consists of the following two parts:

- **AppPackager** — This component packages Business Process Server applications and its dependant components into a single ZIP file. In order to use AppPackager, you need to create an application metadata file, which is an XML file containing the details of all the applications that need to be published. The metadata XML file also contains the information about the application folders that need to be copied.

- **AppDeployer** — This is the application publisher component that publishes the Business Process Server applications on existing Business Process Server servers.

**Important:** If https is enabled, then valid certificate needs to be added to the JDK within BP Server. If valid certificates are not added then publishing of applications and many other operations using WebServices will fail. For more information on how to add the certificate, refer to your Application Server documentation on which BP Server is installed.

Both the AppPackager and AppDeployer components use the Apache ANT build tool that is packaged with Business Process Server or is available as a plug-in with Eclipse in Progress Developer Studio for OpenEdge. This tool uses the `appbuild.xml` file with various targets to package and publish applications.

For BP Server: `appbuild.xml` file is located in the `OEBPS_HOME\apppackager` directory.

For Progress Developer Studio for OpenEdge: `appbuild.xml` file is located in the `DeveloperStudio3.7\eclipse\plugins\com.savvion.studio.resources_<version>_<timestamp>\apppackager` directory.

The AppPackager component uses the application metadata file as the input and packages the applications into a single ZIP file (see Application package details on page 68). The AppDeployer component uses the ZIP file as the input and publishes all the applications. It also copies the resource files specified in the XML metadata file.

Application metadata

You need to create the application metadata XML file containing details of all the Business Process Server applications that need to be packaged and published. The elements of this XML file are controlled by the `ProcessPackage.dtd` file, which is located in the `OEBPS_HOME\conf\resources\common` directory for Business Process Server installation.

The various tags are described below.

- **<Application>** — Defines the entire application package. The name attribute specifies the name to be given to the package zip and the location attribute specifies the location where the file is to be copied. The `<Application>` tag can contain one or more `<Process>` tags and zero or one `<Resources>` tag.

- **<Process>** — The `<Application>` tag has several processes, which are described using the `<Process>` tag. This tag can have zero or one `<Resources>` tag. It has the following four attributes:
  - **Name** — This is a mandatory attribute and is used to specify the name of the process.
• **Type** — This is also a mandatory attribute used to specify the type of process (BP Server, BPM Events, or Web (formerly BPM Workflow)). The acceptable value for this attribute is 'BP Server', 'BPM Events', or 'BPM Workflow.'

**Note:** An application can be a BPM Events application if it has only rules to be published. If the application has a process template as well as rules to be published, then it is a BP Server application.

• **Location** — This optional attribute is used to specify the root directory of the application. If this attribute is not specified, then the default location considered is `OEBPS_HOME\ebmsapps\{Application_Name}` directory in case of BP Server application and `OEBPS_HOME\BPMWebFlow\{Application_Name}` directory in case of Web application.

In case of OpenEdge Developer Studio, the default location would be `Studio_HOME\workspace` for both BP Server and Web applications, where `Studio_HOME` is the Progress Developer Studio for OpenEdge installation directory. The location can be an absolute path or a relative path to the application directory located on the computer where the appPackager tool runs. You can use the variables namely `$oebps.home` or `$sbm.webappdir`, to specify the Business Process Server installation location or the Business Process Server web application directory location.

**Note:** These two variables are resolved with the values specified in the `oebps.conf` file.

• **Operation** — This mandatory attribute is used to specify the operation that needs to be performed when you publish the BP Server, Web and BPM Events applications. The acceptable value for this attribute is 'Reinstall', 'Copy', 'RefreshPt' or 'InstallAsVersion'.

The **Reinstall** value is used in a scenario where a new application needs to be installed, or the existing application needs to be reinstalled, or in case when a new version of the current application needs to be installed. The **Copy** value is used when the user wants to copy all the application-related files (including the process template) on the server. This operation is useful for applications, which are already installed. The **RefreshPt** value is used when the user just wants to reload the process template of the application. The **InstallAsVersion** value is used when you want to publish a versioned process template.

**Note:** If you perform any of these operations on an application that is not installed, then the appDeployer tool installs the application before carrying out the operation.

• **Inheritacl** — This optional attribute is used to specify whether to retain any user permissions in the version of the application you are publishing that are associated with the parent application. The default value is 'false.'

**Note:** If you set the value of Inheritacl attribute to true, then the value of Operation attribute must be set to InstallAsVersion.

• **ParentPt** — If you set the values of Operation attribute to InstallAsVersion and Inheritacl attribute to true, then you must specify the name of the application as the value of ParentPt attribute for the versioned process template.

• **<Resources>** — This tag contains the `<Resource>` tag. A resource can be either a file or a directory. The `<Resource>` tag has the following three mandatory attributes:

  • **src** — This attribute is used to specify the absolute or relative path to a resource. It must be a valid value on the computer where the `appPackager` tool runs.

  • **type** — The acceptable value for this attribute is ‘file’ (for file resource) or ‘dir’ (for directory resource).
target — This attribute is used to specify the valid directory (and path) to which the resource needs to be copied. The path can be an absolute path or a relative path to the directory on the computer where the *appDeployer* tool runs.

**Note:** All resources (both process-specific and application-wide) are copied on the server before the applications are installed.

The target attribute must contain a directory, and not a file.

You can use the ${oebps.home} or ${sbm.webappdir} variables in the 'src' or 'target' attributes to specify the Business Process Server installation location or the Business Process Server web application directory location.

**Note:** These two variables are set to the values specified in the oebps.conf file on the computer where the appPackager and appDeployer tools are run.

---

**Application package details**

Executing the appPackager tool creates a master ZIP file that contains all the applications and the resources specified in the application metadata file. The ZIP file contains three subdirectories namely, BP Server, BPM WebFlow, and BPM Events depending upon the type of the contained applications.

The BP Server subdirectory contains all the applications of type ‘BP Server.’ Similarly, BPM Events and Web (BPM WebFlow) applications are stored in the BPM Events and BPM WebFlow subdirectories respectively.

**Setting properties**

In order to ensure that the appDeployer tool publishes the applications correctly, you need to ensure that valid properties are set in the ant.properties file. This file is located in the OEBPS_HOME\ant directory. The application installer sets the following out-of-box values, which need to be updated if they are not valid.

- **sbmUrl** — Use this property to specify the URL of the portal server. The default value is "http://host.domain.com:<port_number>.

The appDeployer reads Business Process Server credentials from the bmboot.properties file located in the OEBPS_HOME/conf directory.

---

**Working with the AppPackager and AppDeployer Tools**

The appPackager and the appDeployer tools are located in the $Installation_Dir/bin directory where Installation_Dir can be OEBPS_HOME directory or DeveloperStudio3.7\eclipse\plugins\com.savvion.studio.resources_<timestamp>\bin directory, where <timestamp> is the date and time when the build is made.
**appPackager**

To package the applications, you can use the `appPackager.cmd` file (in Windows) or `appPackager.sh` file (in UNIX).

The above command (in Windows) displays the message as follows:

```cmd
appPackager.cmd
[-help]
[applicationMetadataFile]
```

The following table describes these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>-help</td>
<td>Prints this information.</td>
</tr>
<tr>
<td>applicationMetadataFile</td>
<td>Name of the application metadata XML file with complete path.</td>
</tr>
</tbody>
</table>

**appDeployer**

To publish the applications, you can use the `appDeployer.cmd` file (in Windows) or `appDeployer.sh` file (in UNIX).

This command (in Windows) displays the message as follows:

```cmd
appDeployer.cmd
[-help]
[-f appDeployment.xml]
[applicationpackage.zip]
```

The following table describes these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>-help</td>
<td>Prints this information.</td>
</tr>
<tr>
<td>-f appDeployment.xml</td>
<td>This is an optional parameter. You can specify the name of the build file with complete path. If not specified, then the default build file in the <code>OEBFS_HOME\appPackager</code> directory is used.</td>
</tr>
<tr>
<td>applicationpackage.zip</td>
<td>Name of the application ZIP file</td>
</tr>
</tbody>
</table>

**Custom Application publishing**

In certain instances, you may need to execute some specific tasks before or after publishing the Business Process Server applications. In this case, you need to create a custom publishing XML file with appropriate Apache ANT targets.
To do this task:

1. Create a copy of the existing `appbuild.xml` file.

2. Name the file appropriately.

3. Add code to call the custom ANT targets from the ‘deployApp’ target using the `<antcall>` tag.

   If the custom target needs to be called before any of the applications are published, then you must add the `<antcall>` tag for this target before the corresponding `<antcall>` tag for the ‘extractandDeploy’ target. Alternatively, you can add the `<antcall>` tag for this target after the corresponding `<antcall>` tag for the ‘extractandDeploy’ target, if the custom target needs to be called after publishing.
Business Process Server web security

Business Process Server Web Security provides the protection to the Business Process Portal web application from the common vulnerabilities like Cross Site Scripting Requests (XSS) and Cross Site Scripting Request Forgery (XSRF).

BPM portal web security is based on filters taking care of the following tasks:

• Managing XSS attacks by escaping active content from all request parameters.
• XSRF guarding by eliminating requests showing wrong source origin and also validating security token for all system state modifying requests.
• JSHJ vulnerability is addressed by XSRF filter as well as by validating the security token for all requests returning JavaScript easy parseable data as JSON.
• Additional level security is achieved by enabling HTTP only session cookie and making XSS attack more complicated.

Note: To activate POST HTTP request for sensitive parameters, in bpmportal.conf file located in OEBPS_HOME\conf folder set the bpmportal.post.sensitive.parameters=true, the default value for this parameter is false. This enables POST HTTP requests for sensitive parameters passed from BP Server to BPM WebFlow or from BPM WebFlow to another BPM WebFlow Subprocess.

For details, see the following topics:

• Configuring web security
• XSS handler implementation
• XSRF handler implementation
• **Cross domain security**

# Configuring web security

Web security configuration is provided in the `bmwebsecurity.conf` file. This file is available at `OEBS_HOME/conf` directory. Using this file you can enable or disable the individual components for configuring web security.

The following parameters are available in the `bmwebsecurity.conf` file.

<table>
<thead>
<tr>
<th>Table 17: Configuring web security</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>websecurity.logger.factory</td>
</tr>
<tr>
<td>websecurity.xss.enabled</td>
</tr>
<tr>
<td>websecurity.xss.trace</td>
</tr>
<tr>
<td>websecurity.xss.handler.config.defaultconfig</td>
</tr>
<tr>
<td>websecurity.xss.handler.config.default.handlerfactory</td>
</tr>
<tr>
<td>websecurity.xss.handler.config.default.configfile</td>
</tr>
</tbody>
</table>
## Configuring web security

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>websecurity.xsrf.enabled</td>
<td>Enables the XSRF checking when set to <code>true</code></td>
</tr>
<tr>
<td>websecurity.xsrf.trace</td>
<td>Enables DEBUG logging into <code>bmwebsecurity.log</code> along with info messages, when set to <code>true</code></td>
</tr>
</tbody>
</table>
| websecurity.xsrf.handler.defaultconfig | The default xsrf configuration to use when no configuration is explicitly provided to the xsrf filter. The configuration can be defined using the prefix `websecurity.xsrf.handler.config.<config-name>`.
| websecurity.xsrf.handler.config.default.handlerfactory | The implementation class name of the `com.savvion.sbm.websecurity.xsrf.impl.DefaultXSRFHandlerFactory` configuration `default`. This parameter allows you to redefine XSRF filter business implementation. Handler mechanism is similar to XSS filter. Interfaces names are `IXSRFHandler` for handler and `IXSRFHandlerFactory` for its factory. Package name is `com.savvion.sbm.websecurity.xsrf.api`. **Note:** It is recommended not to modify the default handler values. |
| websecurity.xsrf.handler.config.default.configfile | The XSRF config file location for default configuration. |

The following parameters are available in the `bpmportal.conf` file. This file is available at `OEBPS_HOME/conf` directory.

### Table 18: Configuring web security in Portal

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oebps.front.host</td>
<td>Value for this parameter is a comma separated list of hosts which are exposed to access the system. See Setting up oebps.front.host parameter on page 75 for more information. This parameter value has to be changed after the installation of the system. <strong>Important:</strong> This parameter does not appear in config file by default. You have to add this while enabling ref filter.</td>
</tr>
<tr>
<td>bpmportal.post.sensitive.parameters</td>
<td>If set to true, this enables POST HTTP requests for sensitive parameters passed from BP Server to BPM WebFlow or from BPM WebFlow to another BPM WebFlow Subprocess. The default value for this parameter is false.</td>
</tr>
</tbody>
</table>
Logger factory class name

This configuration provides the implementation class name of the
com.savvion.sbm.websecurity.common.api.IWebSecurityLoggerFactory interface. This logger
factory is used to log all web services messages to the log file. The default logger factory implementation uses
the BP Server logger which logs the messages to the bmwebsecurity.log file located under
OEBPS_HOME/logs directory.

The default logger configuration file bmwebsecuritylog.conf can be found under the OEBPS_HOME/conf
directory.

The default logging mechanism can be changed by providing an alternate implementation for the interface
com.savvion.sbm.websecurity.common.api.IWebSecurityLoggerFactory and setting the
implementation class name to the websecurity.logger.factory property in bmwebsecurity.conf file
as shown below.

websecurity.logger.factory=com.savvion.sbm.websecurity.common.impl.
DefaultWebSecurityLoggerFactory

XSS configuration for web security

This section explains XSS related configuration for web security.

Enabling/Disabling XSS feature

Using websecurity.xss.enabled parameter, you can enable the xss feature in the portal server. For that
you have to set this parameter to true in bmwebsecurity.conf file. By default this functionality is not
enabled.

Enabling/Disabling tracing for XSS

Using websecurity.xss.trace parameter, you can enable the tracing for XSS component. For that you
have to set this parameter to true in bmwebsecurity.conf file. By default this functionality is not enabled.
So only the information messages are logged into bmwebsecurity.log file.

XSS config handler configuration

To support for multiple XSS Handlers’ configurations in a single JVM, this web security configuration provides
a facility to have multiple handlers’ configuration defined with its own unique configuration file.

To define a new configuration, perform the following:

1. Provide unique name for each configuration handler.
2. Set the name of the handler to the XSS filter in the web.xml.

Important: The XSSFilter uses the default configuration, if no value is passed to it, on which configuration
to use.

XSS config new handler configuration

The default configuration property is websecurity.xss.handler.defaultconfig=default.
To define a new handler configuration:

1. Prefix all the properties to the handler configuration with `websecurity.xss.handler.config.<configname>.propertyName=propertyValue`.

2. Each handler configuration must have the following two entries.
   a) `websecurity.xss.handler.config.default.handlerfactory`
      This property contains the name of the implementation class for `com.savvion.sbm.websecurity.xss.api.IXSSHandlerFactory` interface.
   
   b) `websecurity.xss.handler.config.default.configfile`
      This property contains the name of the config file. The configured file must be in the classpath.
      The default configuration file with the name `bmxssconfig.xml` is available under `OEBPS_HOME/conf` directory.

---

XSRF configuration for web security

This section explains XSRF related configuration for web security.

Enabling/Disabling XSRF feature

Using `websecurity.xsrf.enabled` parameter, you can enable the XSRF feature in the portal server. For that you have to set this parameter to `true` in `bmwebsecurity.conf` file. By default this functionality is not enabled.

Enabling/Disabling tracing for XSRF

Using `websecurity.xsrf.trace` parameter, you can enable the tracing for XSRF component. For that you have to set this parameter to `true` in `bmwebsecurity.conf` file. By default this functionality is not enabled and only the information messages are logged into `bmwebsecurity.log` file.

Setting up `oebps.front.host` parameter

Value for this parameter is a comma separated list of hosts which are exposed to access the system. Not specifying this parameter or keeping it empty will disable corresponding RefXSRF filter.

Here are some recommendations to setup `oebps.front.host` parameter.

- Specify localhost in the list recommended for development instances.
- Specify long and short notation (when base domain details omitted) for production systems.

For example, if BPM machine name is `bpm`, and the domain name is `visa.com` then the `bpm`, and `bpm.visa.com` entries should be specified.

XSRF config handler configuration

XSRF supports multiple handlers' configurations with each handler having its own configuration file.

The default handler configuration, the below property is used, if there is no handler configured while integrating with the portal.

`websecurity.xsrf.handler.defaultconfig=default`

To define a new configuration:
1. Prefix all the properties to the configuration with
   
   ```
   websecurity.xsrf.handler.config.<configname>.propertyName=propertyValue.
   ```
   
   All the properties configured with the associated handler config name is accessed using the API.

2. Each handler configuration must have at least the following two entries.
   
   a) `websecurity.xsrf.handler.config.default.handlerfactory`

   This property contains the name of the implementation class for
   
   ```
   com.savvion.sbm.websecurity.xsrf.api.IXSRFHandlerFactory interface.
   ```
   
   b) `websecurity.xsrf.handler.config.default.configfile`

   **Caution:** This property contains the name of the config file. The configured file must be in the classpath.

The default XSRF configuration can be found in `bmxsrfconfig.xml` under `OEBPS_HOME/conf` directory.

**XSS handler implementation**

When enabled, the default implementation of the XSS handler sanitizes the incoming data before accepting it. Sanitization process is different for different types of data such as HTML text and plain text. To identify whether the incoming data is HTML text or plain text, it uses the `Element Detection Pattern` regular expression as explained in the following sections.

**Sanitizing HTML text**

The default implementation uses the Tag White list approach in which all accepted tags along with accepted attributes are taken from the configuration file. Only those tags that are meant to be accepted in the configuration file are accepted. The action on all unknown or unaccepted tags are taken based on the configuration set as explained in XSS actions on page 77.

**Sanitizing plain text**

The plain text is sanitized by escaping all the unsafe HTML characters such as ‘>’, ‘<’ and ‘”’ characters using the Apache StringEscapeUtils. You can change this behavior by modifying the `unmatched-content-type-action` attribute for the tag `Element-detection-pattern` under the XSS conditions.

**XSS configuration**

XSS configuration is stored in `bmxssconfig.xml` by default. It is available under `OEBPS_HOME/conf` directory. Each entry in the xss configuration is explained in the following sections. The XSS configuration is divided into four parts. They are:

- XSS Conditions
- XSS Actions
XSS conditions

The config entry xss-conditions are used to determine whether the input data is plain text or html text. The conditions are applied against the input data, if the condition matches, then the input data is considered as html text and sanitized according to the HTML sanitization process. If it does not match, then it uses the action configured for the unmatched-content-action-type attribute.

The default xss-conditions snippet:

```xml
<xss-conditions>
  <element-detection-pattern unmatched-content-action-type='HTML_ESCAPE'>
    (.*<([^\s].*)>&.*)
  </element-detection-pattern>
</xss-conditions>
```

Element detection pattern

The default xss-conditions look for possible element patterns such as any string starting with "<" symbol. If the pattern is matched against the input data, then the data is considered HTML text and it is passed to the xss handler for sanitization. If there are no element pattern found in the given data, then it considers it as plain text and uses the unmatched-content-action-type attribute to take further actions on that data.

Note: The value of the element-detection-pattern should be a valid java regular expression.

Unmatched content action type

If there is no element pattern found in the input data, then the xss handler takes one of the following actions on the input data before returning.

HTML_ESCAPE – Escapes the characters in the input data using HTML entities.

Example:

"bread" & "butter"

becomes:

&quot;bread&quot; &quot;butter&quot;.

NO_ACTION

No action is taken on the input data and it is accepted as it is.

One of the above values should be configured for the attribute named unmatched-content-action-type under the <element-detection-pattern> element.

Note: By default, it escapes the html entities in the input data.

XSS actions

The XSS handler applies the xss rules on the input data, which are configured under the <xss-rules> sections in the bmxxssconfig.xml.

As part of the input data validation, the following things can happen.
1. Finding tags that are not accepted.
2. Finding attributes that are not accepted.

**XSS rules** on page 79 section explains about how to define a tag attribute and whether to allow or stop a tag attribute.

**Action on unaccepted attributes** on page 78 and **Action on unaccepted tags** on page 78 explains what action should be taken when an attribute or tag is not accepted.

A sample snippet is given below:

```xml
<xss-actions>
  <unaccepted-tag-action-type>ENCODE</unaccepted-tag-action-type>
  <unaccepted-attribute-action-type enabled='true'>REMOVE</unaccepted-attribute-action-type>
</xss-actions>
```

**Action on unaccepted tags**

This configuration specifies what action to take on the tags that are not accepted in the user data.

---

**Note:** All the outputs given under each section are based on the input data. (\<b\>This is bold text \<script\>alert("hi")\</script\></\>\) where the script tag is not allowed.

---

**The possible values are:**

- **FILTE** — Removes the tag, but keeps the content.
  
  Output: \<b\> This is bold text alert("hi") \</b\>

- **REMOVE** — Removes tag along with the content in it.
  
  Output: \<b\> This is bold text \</b\>

- **ENCODE** — Encodes the tag, so that the tag with the content is displayed, but not executed when rendered in the browser.
  
  Output: \<b\> This is bold text &lt;script&gt;alert(&quot;hi&quot;)&lt;/script&gt;</b>

- **THROW_EXCEPTION** — Throws exceptions to the user.
  
  It throws an exception to the user. This is not recommended.

- **APPLY_CDATA** — Applies the content with the CDATA tag.
  
  Output: \<b\> This is bold text <![CDATA[ &lt;script&gt;alert("hi")&lt;/script&gt;]]\>

Based on the needs of your application security levels, you can decide what option to use for your application.

**Action on unaccepted attributes**

This configuration specifies what action should be taken on the attributes that are not accepted in the user data. The possible values could be:

- **REMOVE** — Removes the attribute and its value.
- **THROW_EXCEPTION** — Throws exception to the user. This option is not recommended.
- **ENCODE_VALUE** — Encodes the attribute value with base64.

The default value of **unaccepted-attribute-action-type** is **REMOVE**.
XSS rules

The xss rules configuration provides details about what tags and attributes are accepted and rejected in the input data.

Import rules from different files

The xss-rules can be written in multiple files and can be imported to the main configuration file using the <import> tag and the file name as the name of the file in the classpath, and name of the xss-rules section in that imported file. Each imported file has the xss-rules same as those explained in this section.

An example would be:

```xml
<import file='resources/conf/bmwebsecurity/bmxsshtml4rules.xml' name="xss-default-html4"/>
```

The above tag imports the xss-rules named as "xss-default-html4" from the above file location in the classpath.

Accept common attributes

This section explains about common attributes:

The following code explains configuration about which attributes should be accepted for any tag.

Example:

```xml
<accept-common-attr name="align">
</accept-common-attr>
```

This accepts the align attribute in any tag.

This can be further restricted by using the <except-tag> which restricts the common attribute.

Example:

```xml
<accept-common-attr name="id">
  <except-tag name="script" />
</accept-common-attr>
```

This indicates that the "id" attribute can be accepted for any tag except the script tag.

The value of the attribute can be restricted with the multiple regular expressions given below. If there are no regular expressions defined, then the attribute is accepted irrespective of its value.

```xml
<accept-common-attr name="align">
  <regexp>center|left|right</regexp>
</accept-common-attr>
```

The above configuration indicates that the align attribute should be accepted for any tag, but the value of the attribute must be center, left or right only.

**Note:** The value of the <regexp> must be a valid java regular expression.

Accept tags

Which tags should be accepted in configuration for input data is defined using <accept-tag>.

For example:

```xml
<accept-tag name="b">
</accept-tag>
```
The above rule says that the "b" bold tag can be accepted.

Tag specific attribute: Each accept tag does allow multiple attributes that can be allowed for that tag only is given below.

```xml
<accept-tag name="b">
  <attr name="align">
    <regexp>center|left|right</regexp>
  </attr>
  <attr name="style" /> 
</accept-tag>
```

The above rule indicates that the "b" tag is accepted with the align and style attributes within it.

**Reject common attributes**

The attribute configured with the following tag is rejected for all tags.

```xml
<reject-common-attr name='style' />
```

**Reject tags**

The tags configured with the following rule is rejected in the input data.

```xml
<reject-tag name='script' />
```

**Reject attributes for a tag**

The following snippet rejects the attribute for the given tag.

```xml
<reject-tag name="a">
  <attr name="href" />
</reject-tag>
```

The default configuration will not allow the following tags:

```xml
<reject-tag name="SCRIPT" />
<reject-tag name="NOSCRIP" />
<reject-tag name="IFRAME" />
<reject-tag name="FRAMESET" />
<reject-tag name="IMG" />
<reject-tag name="META" />
<reject-tag name="EMBED" />
<reject-tag name="OBJECT" />
<reject-tag name="HEAD" />
<reject-tag name="LINK" />
```

**Note:** Go through the following files for more information on what tags and what attributes are accepted or rejected. conf\resources\conf\bmwebsecurity\bmxsshtml4rules.xml 
conf\resources\conf\bmwebsecurity\xss-html4-standard-rules.xml

**Important:** The recommended approach is to add accept or reject rules through the bmxssconfig.xml rather than modifying the above files.

**Exclude URLs**

This exclude url's list is used to list all the urls that must be excluded from checking for the XSS attacks.
URL

- Each URL should be relative to the context path and must start with "/".
- The individual url-patterns use the ANT style patterns for pattern matching.

By default all the urls that have the extensions namely, CSS, JS, PNG, GIF and SWF are excluded.

```xml
<exclude-url-patterns>
    <url-pattern>/**/*.css</url-pattern>
    <url-pattern>/**/*.js</url-pattern>
    <url-pattern>/**/*.png</url-pattern>
    <url-pattern>/**/*.gif</url-pattern>
    <url-pattern>/**/*.swf</url-pattern>
</exclude-url-patterns>
```

**XSRF handler implementation**

BP Server provides the default xsrf handler implementation using the class. This implementation can be replaced with the configuration exposed in the file.

This default implementation generates an XSRF token (key, value pair) under each html form element that the portal server generates. The XSRF token is expected whenever a user submits the form from the browser. Upon successful evaluation of the XSRF token, the request is accepted for further processing by the portal server.

If there is no token found in the request, or an invalid token found in the request, then that user's session is invalidated, and forwarded to the login page with the message stating that "For security reasons, your session has been terminated. Please login again or contact administrator."

**Note:** The default XSRF implementation is applied for POST requests only. It does not evaluate the GET Requests and AJAX requests.

**XSRF configuration**

The following configuration options are provided in the XSRF configuration.

**Redirect URL on possible attack**

When a possible XSRF attack is observed that is when there is no valid xsrf token found in the request, Business Process Portal does not invalidate the session automatically. BPM portal forwards the user to the configured target url when it finds a possible xsrf attack.

The default implementation of this target url invalidates the current user session and gives a message to the user saying "For security reasons, your session has been terminated. Please login again or contact administrator."

The target url should be relative to the context path of the application.

```xml
<redirect-url-on-attack>/bpmportal/xsrf_handler.jsp</redirect-url-on-attack>
```
User agent check

If `enable-user-agent-check` is set to `true`, then Business Process Portal stores the Http User Agent information in the user's session. As long as the user session is valid, it expects that all the requests from that user should come from the same User Agent. It checks against each request to see if the User Agent is matching, if not it treats the request as invalid and redirects the user to the url configured for redirect-url-on-attack.

```xml
<enable-user-agent-check>false</enable-user-agent-check>
```

The default value is false.

Remote address check

If `enable-user-remote-address-check` is set to `true`, then Business Process Portal stores the user's remote address in that user's session. As long as the user session is valid, it expects that all the further requests from that user should come from the same remote address. It checks against each request to see if the request is coming from the same remote address, if not it treats the request as invalid and redirects the user to the url configured for redirect-url-on-attack.

```xml
<enable-user-remote-address-check>false</enable-user-remote-address-check>
```

The default value is false.

Exclude URLs for XSRF attacks

This exclude url's list is used to list all the urls that must be excluded from checking for the XSRF attacks. Each URL should be relative to the context path and must start with "/".

The individual url-patterns uses the ANT style patterns for pattern matching.

**Note:** By default all the urls that have the extensions namely CSS, JS, PNG, GIF and SWF are excluded.

Currently the following urls are excluded from checking for XSRF attacks.

```xml
<exclude-url-patterns>
  <!-Excluding the jsp pages under the bpmportal directory directly, but not the sub directories in it.-->
  <url-pattern>/bpmportal/*.jsp</url-pattern>
  <url-pattern>/**/*.css</url-pattern>
  <url-pattern>/**/*.js</url-pattern>
  <url-pattern>/**/*.png</url-pattern>
  <url-pattern>/**/*.gif</url-pattern>
  <url-pattern>/**/*.swf</url-pattern>
</exclude-url-patterns>
```

Disable double session cookie

XSRF filter uses double session cookie approach to identify and validate source of requests. This feature can be disabled in some cases, for example when no custom JSP is used.

The following tag has to be added in `bmxsrfconfig.xml`.

```xml
<disable-token-cookie>true</disable-token-cookie>
```
Cross domain security

The crossDomainSessionSecurity parameter is available in web.xml file in OEBPS_HOME\webapps\deploy\sbm\WEB-INF folder.

The default value of this parameter is set to false. This parameter needs to be set to true to enable web security (besides other web security flags) in Business Process Server. If this flag is set to true, then pages with controls and DB adaplets cannot be accessed from outside Business Process Portal.
Glossary

**ACL manager**
In Business Process Server, Access Control List Manager provides a finer, more precise control over user access rights for resources and actions.

**Activity workstep**
In Business Process, the basic unit of work; must be performed by one or more human performers (valid individual user, multiple users or user group).

**Adapter**
A Java class that integrates remote, third party classes and actions with Business Process. An adapter can automate certain functions and tasks performed by a remote server or other external systems.

**Administration**
A module in Business Process Portal enabling the administrator to perform tasks such as installing/uninstalling applications, modifying configuration parameters controlling Business Process operations, and manage users, groups and access control. The Administration module is visible only to application users who have permissions to access it.

**Application**
In Business Process, an application is an installed, executable business process that automates a business flow.

**Balanced scorecard**
A management application in the Management module that measures performance by analyzing how an organization’s business activities help it achieve its strategic goals. The Balanced Scorecard provides an analysis from a range of perspectives.

**BAM**
Business Activity Management combines Business process management with strategic and analytical information on specific business performance indicators, providing real-time status information and identifying critical events to assist senior management in making informed business decisions.
BPM Events
A Business Process Server component that provides an open event-driven rule engine to formulate and enforce policies in business applications.

BPM Webflow
A Business Process Server component that enables users to develop customizable, sophisticated presentation flows for business processes, install them as Web applications, and execute them on their Web browsers.

BPEL
BPEL (Business Process Execution Language) for Web services is an XML-based language designed to enable task-sharing for a distributed computing or grid computing environment - including across multiple organizations - using a combination of Web services.

BPMN
BPMN (Business Process Modelling Notation) provides businesses with the capability of defining and understanding their internal and external business procedures through a Business Process Diagram giving organizations the ability to communicate these procedures in a standard manner.

BP Server
A Business Process Server component that provides a flexible, lightweight, scalable workflow process engine for intranets, extranets, and the Internet.

Business calendar
A Business Process Server feature that accurately calculates the Due Date of tasks, and provides support for multiple business calendars across different time zones.

Business flow
The logical sequence of process activities, related to one another by a triggering activity, to achieve an outcome. It represents a business process that begins with a commitment and ends with the termination of that commitment. In Business Process Server, business flow includes Workflow (the flow of all human-performed activities), integration flow (the flow of activities performed by systems) and presentation flow (from a user’s viewpoint, the flow of data from one Web page to the next).

Business logic
The control flow and information flow among worksteps that define a business process.

Business object
A representation of an activity in the business domain, including its name, definition, attributes, behavior, relationships and constraints.

Business process
A process involving multiple worksteps in the form of operations, interactions and notifications performed by a user, group of users, an external adapter, or a script.

Business Process Server application
An application is an implementation of a business process. It can contain one or more process templates, performers, adapters, customized forms or rules. An application can be published, installed and run on BP Servers. In Business Process Server, an application is an installed, executable business process that automates a Workflow.
**Business Process Server Web services**
A Business Process Server component that allows application developers to: a) publish their applications as Web services, and b) find and convert other available Web services on the Internet into Business Process Server applications.

**Business Process Portal**
A Business Process Server component that offers users, managers, administrators and developers a unified, customizable portal for single sign-on access to all Business Process Server functionalities to which they are granted permission.

**Business process management**
The concept of guiding work activities through a multi-step business process in order to improve performance and reduce costs within and across functional business units.

**Business Process Modeler**
A stand-alone component that enables users to design templates for basic business processes.

**Business rule**
A combination of elements, including validation edits, logon verifications, database lookups, policies and transformations, that represent an enterprise’s way of doing business.

**Control flow**
The sequences of worksteps and workstep conditions, as defined in a process template in Progress Developer Studio for OpenEdge or Business Process Modeler.

**Dashboard**
A Business Process Server feature that provides a graphic overview of the status of several business processes on a single Web page, enabling users to monitor the progress of each process. Users can view business processes across all applications or for a selected application.

**Dataslot**
A data placeholder that persists through the entire process and defines the information flow of the business process. Dataslots are associated with processes, where they can add information into (Input type) or out of (Output type) worksteps, and appear as editable or read-only fields on a user’s interface.

**Expression editor**
A Business Process Server tool that enables users to define complex conditional expressions within a Decision gateway to support their business requirements.

**Group**
In Business Process Server, an entity that has as members valid users or other groups who perform related work and have authorized access to specific components.

**Heatmap**
A Business Process Server feature that provides a convenient, graphical tool for managers to visually locate the bottlenecks in the process execution. It helps managers to get an overview of the status of the currently active instances, identify suspended instances, and analyze the history of the completed instances.

**Home**
A module in Business Process Portal through which users interact with Business Process Server. Using the Home module, users complete entries to various tasks and applications, update profile, set preferences, and link to the support infrastructure required to achieve these tasks. The Home module is the primary interface for application users.
Infopad

In Business Process Server, a data structure used to capture business metrics, typically displayed as a table with one or two dimensions.

Instance

An individual object within a specific class. In Business Process Server, a self-contained unit that is created each time you use a process template to run a Business Process Server application.

KPI

Key Performance Indicator, used in the Balanced Scorecardsystem, that provides the data translating enterprise goals into a set of measurable objectives.

Managed Adapter

In Business Process Server, a Managed Adapter is an implementation of an adapter interface that facilitates data exchange between Business Process Server processes and external applications.

Management

A module in Business Process Portal enabling the managers to query, report, and control processes and resources for application users. The Management module is visible only to application users who have permissions to access it.

Migration

The process of moving from the use of one operating environment to another operating environment that is typically seen as improvement. Migration can involve moving to new hardware, new software, or both. It may involve a new application, another type of database, or a redesigned network. Migration is also used to refer simply to the process of moving data from one storage device to another. Business Process Server supports data migration as well as application migration.

Performer

An entity that executes a workstep. Depending on the workstep type, the performer can be a human user, a group of users, an adapter or other external performer, or a script.

Presentation flow

The flow of information and user input from one interface to the next. Typically related to a single Activity workstep in the process and generated in a BPM Webflow environment.

Process engine

Orchestrates the execution of business processes and also coordinates conversations among process engines based on public processes, which forms the backbone of global business collaboration.

Process refresh

A Business Process Server feature for replacing the installed process without versioning, facilitating the running process instances to refresh and seamlessly adapt to the new Workflow.

Process template

In Business Process Server, a model of business flow that includes worksteps, connectors and dataslots. After users publish and install it as an application in Business Process Server folder structure, they can use the application to create process instances.

Progress Developer Studio for OpenEdge

An Integrated Development Environment for Business Process Server that enables application users to develop and publish a Business Process Server application without leaving the development environment.
Role
The actions and activities assigned to a valid application user who is a member of a group. In Business Process Server, only members of a group can be assigned a role. A role indicates the relationships of the user in a group context.

Rollback
In Business Process Server, a feature that restarts the Workflow from a workstep previously selected as the rollback point in the process, performed automatically in the event of a failure.

Rule wizard
An interactive utility that enables application users to quickly develop rules that can be applied to a business process.

Swim lanes
Used in Workflow diagrams to organize complex processes across functional boundaries. For example, seen as horizontal lines on a process map, swim lanes can be used to place individual task steps into different categories that depend on task ownership.

Task
In Business Process Server, a performer is assigned one or more work items that the performer sees as tasks. There are two types of tasks: Assigned, which are assigned specifically to you; and Available, which are available to be performed by you or other members of your user group.

User
In Business Process Server, a valid human performer with authorized access to specific modules.

Workflow
The logical sequence of activities performed by human performers. Workflow includes the tasks, procedural steps, organizations or people involved, required input and output information, and tools needed for each activity in a business process.