



**FUJITSU Software  
Interstage Business Process  
Manager  
V11.4.1**

**Server Administration Guide**

J2U3-0057-11ENZ0(00)  
August 2017

Publication Date	August 2017
Revision	11
Trademarks	<p>Interstage is a trademark of Fujitsu Limited.</p> <p>Microsoft, Windows, and Windows Server are registered trademarks of Microsoft Corporation in the United States and other countries.</p> <p>UNIX is a registered trademark of the Open Group in the United States and in other countries.</p> <p>Oracle and Java are registered trademarks of Oracle Corporation and its subsidiary and related companies in the United States and other countries.</p> <p>Linux is a registered trademark of Linus Torvalds in the USA and other countries.</p> <p>Red Hat, the Red Hat "Shadow Man" logo, RPM, Maximum RPM, the RPM logo, Linux Library, PowerTools, Linux Undercover, RHmember, RHmember More, Rough Cuts, Rawhide and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.</p> <p>All other trademarks and trade names mentioned have been registered by their respective manufacturer.</p>

All Rights Reserved,  
 Copyright © FUJITSU  
 LIMITED 2005-2017

All rights reserved, including those of translation into other languages. No part of this manual may be reproduced in any form whatsoever by means of photocopying, microfilming or any other process without the written permission of Fujitsu Limited.

### **High Risk Activity**

The Customer acknowledges and agrees that the Product is designed, developed and manufactured as contemplated for general use, including without limitation, general office use, personal use, household use, and ordinary industrial use, but is not designed, developed and manufactured as contemplated for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (hereinafter "High Safety Required Use"), including without limitation, nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system. The Customer shall not use the Product without securing the sufficient safety required for the High Safety Required Use. In addition, Fujitsu (or other affiliate's name) shall not be liable against the Customer and/or any third party for any claims or damages arising in connection with the High Safety Required Use of the Product.

---

## Table of Contents

	<b>About this Manual.....</b>	<b>9</b>
<b>1</b>	<b>Architecture Overview.....</b>	<b>12</b>
1.1	Configuration Overview.....	12
1.2	System Architecture.....	13
1.2.1	Interstage BPM Server Tier.....	13
1.2.2	Interstage BPM Web and Client Tier.....	17
1.3	UDDI Repository.....	17
1.4	Interstage BPM as a Service (SaaS).....	18
1.5	Interstage BPM Server Self-Healing.....	19
<b>2</b>	<b>Stopping and Starting Interstage BPM Server.....</b>	<b>20</b>
2.1	Stopping and Starting Interstage BPM for WebLogic.....	20
2.1.1	Stopping Interstage BPM for WebLogic.....	20
2.1.2	Starting Interstage BPM for WebLogic.....	21
2.2	Stopping and Starting Interstage BPM for JBoss.....	22
2.2.1	Stopping Interstage BPM for JBoss.....	23
2.2.2	Starting Interstage BPM for JBoss.....	23
<b>3</b>	<b>Configuring Interstage BPM Server.....</b>	<b>24</b>
3.1	Configuring the Email Listener.....	24
3.2	Changing Parameter Values Using Tenant Management Console.....	24
3.3	Using the Configuration Export and Import Scripts.....	25
3.4	Changing the Database and Directory Service Configuration.....	28
3.5	Configuring the Document Management System (DMS).....	30
3.6	Integrating Interstage BPM with Other Servers.....	32
3.6.1	Integration of Interstage BPM to Interstage BPM.....	33
3.6.2	Setting a Different Interstage BPM Linkage User.....	34
3.7	Configuring Business Calendars.....	35
3.7.1	Business Calendar Format.....	35
3.7.2	Shift Example.....	39
3.8	Configuring Log Files.....	39

---

3.8.1	System-level log files and Tenant-specific log files.....	39
3.8.2	Status log files.....	40
<b>3.9</b>	<b>Dynamically Reloadable Properties.....</b>	<b>41</b>
<b>3.10</b>	<b>Changing Tenant Resource Permissions.....</b>	<b>42</b>
<b>4</b>	<b>System Configuration Parameters of the Interstage BPM Server.....</b>	<b>44</b>
4.1	AllowNonDMSFileAttachmentsInEmailAction.....	44
4.2	AnalyticIntegrationSuspendHours.....	44
4.3	ApplicationSecurityMode.....	45
4.4	AuditLogRecycleSize.....	45
4.5	ClusterAutomateFailoverEnabled.....	45
4.6	ClusterHeartbeatInterval.....	46
4.7	ClusterInterceptEvents.....	46
4.8	ClusterMulticastPort.....	46
4.9	ClusterMulticastTopic.....	47
4.10	DBSQLQueryTimeout.....	47
4.11	EncryptionKey.....	47
4.12	ErrorLogLevel.....	47
4.13	ErrorLogRecycleSize.....	48
4.14	HealthCheckInterval.....	48
4.15	JMSNamingProviderURL.<Hostname>.....	48
4.16	JSExtCmdsClasses.....	49
4.17	JSExtCmdsFile.<Hostname>.....	49
4.18	KPIListenerConfigFile.<Hostname>.....	49
4.19	LogFileRecycleNumber.....	49
4.20	MaxProcessDefinitionCacheSize.....	50
4.21	MaxProcessInstanceCacheSize.....	50
4.22	ObjectCacheEnabled.....	51
4.23	ProactiveNotificationEnabled.....	51

---

---

4.24	ProactiveSQNotificationEnabled .....	51
4.25	SearchIndexUpdateInterval.....	51
4.26	ServerBaseURL.....	52
4.27	ServerEmailBaseURL.....	52
4.28	ServerGNEmailCustomizerClass.....	52
4.29	ServerHost.<Hostname>.....	53
4.30	ServerLogsDirectory.....	53
4.31	ServerMaxUnhandledEvents.....	53
4.32	ServerName.....	53
4.33	ServerRoot.<Hostname>.....	54
4.34	ServerSharedRoot.<Hostname>.....	54
4.35	SMTPPassword.....	54
4.36	SMTPServerHost.....	54
4.37	SMTPServerPort.....	54
4.38	SMTPUserName.....	55
4.39	SSOToken.....	55
4.40	SupportGroupWorkItem.....	55
4.41	SupportLogBufferSize.....	55
4.42	SupportLogRecycleSize.....	56
4.43	SupportLogThreshold.....	56
4.44	SuspendEventWhenProcessInError.....	56
4.45	TraceLogLevel.....	57
4.46	TraceLogRecycleSize.....	57
4.47	WSCallTimeOut .....	57
4.48	WSHttpProxyHost.....	58
4.49	WSHttpProxyPort.....	58
5	<b>Properties of Interstage BPM Server Tenants.....</b>	<b>59</b>
5.1	AdminRole.....	59

---

---

5.2	AuditLogRecycleSize.....	59
5.3	DDFrameWorkImplClass.....	60
5.4	DirectoryImplClass.....	60
5.5	DirectoryService.....	60
5.6	EmailListenerAutoReplyEnabled.....	60
5.7	EmailListenerDeleteInvalidMessages.....	60
5.8	EmailListenerEmailAddress.....	61
5.9	EmailListenerEnabled.....	61
5.10	EmailListenerPassword.....	61
5.11	EmailListenerPollingInterval.....	61
5.12	EmailListenerPOPPort.....	61
5.13	EmailListenerPropertiesFile.....	61
5.14	EmailListenerServerHost.....	61
5.15	EmailListenerUserName.....	62
5.16	EmailNotificationEnabled.....	62
5.17	EmailStyleSheetFile.....	62
5.18	ErrorLogLevel.....	62
5.19	ErrorLogRecycleSize.....	62
5.20	IASSOLoginConfFileEntryName.....	62
5.21	IBPMUserStoreActive.....	62
5.22	LDAPAccessUserID.....	63
5.23	LDAPAccessUserPassword.....	63
5.24	LDAPContextFactory.....	63
5.25	LDAPDBSuffix.....	63
5.26	LDAPFilter.....	64
5.27	LDAPGroupCacheAgeSec.....	64
5.28	LDAPGroups.....	64
5.29	LDAPGroupsDN.....	64

---

---

5.30	LDAPPeople.....	64
5.31	LDAPServer.....	64
5.32	LDAPuid.....	65
5.33	LDAPUseDefUsersGroups.....	65
5.34	LDAPUsersDN.....	65
5.35	LogFileRecycleNumber.....	66
5.36	ManagerAttribute.....	66
5.37	MetadataRepositoryPassword.....	66
5.38	MetadataRepositoryURL.....	66
5.39	MetadataRepositoryUserName.....	66
5.40	RequiredUserAttribute.....	67
5.41	SecuritySwitch.....	68
5.42	SendAnalyticEvents.....	69
5.43	ServerEmailAddress.....	69
5.44	ServerEmailClientPage.....	69
5.45	ServerLogsDirectory.....	69
5.46	ServerNotificationUserProfileField.....	69
5.47	ServerPassword.....	70
5.48	ServerReassignMode.....	70
5.49	ServerUserName.....	70
5.50	SSOAuthentication.....	71
5.51	SWAPLinkagePassword.....	72
5.52	SWAPLinkageUserName.....	72
5.53	TeamAttribute.....	72
5.54	TemplatePublishers.....	73
5.55	TraceLogLevel.....	73
5.56	TraceLogRecycleSize.....	73
5.57	UDDIBusinessName.....	73

---

---

5.58	UDDIInquiryURL.....	73
5.59	UDDIPublisherPassword.....	73
5.60	UDDIPublisherUserName.....	74
5.61	UDDIPublishURL.....	74
5.62	WebDAVKeyStore.....	74
5.63	WIResponsible.....	74
<b>6</b>	<b>Making a Backup of Your Data.....</b>	<b>75</b>
6.1	Configuration Files.....	75
6.2	Log Files.....	75
6.3	DMS Directory.....	75
6.4	Database.....	75
	<b>Appendix A Database Schema.....</b>	<b>76</b>
	<b>Appendix B Message List of Interstage BPM Server.....</b>	<b>109</b>
	<b>Appendix C Troubleshooting.....</b>	<b>116</b>
C.1	Log File Information.....	116
C.2	Resolving Specific Error Situations.....	116
C.2.1	Interstage BPM Server Fails to Start.....	116
C.2.2	Error in Trace.log.....	117
C.2.3	Timeout During JavaScript Execution.....	117
C.2.4	Failure in Writing to an Oracle Database.....	117
C.3	Errors during Setup of the Interstage BPM Server .....	118
C.4	Errors Related to Interstage BPM Database Creation/Update.....	121
C.5	Contacting Your Local Fujitsu Support Organization.....	122
	<b>Glossary .....</b>	<b>124</b>
	<b>Index .....</b>	<b>130</b>

---



## About this Manual

This manual describes how to configure and administrate Interstage Business Process Manager (Interstage BPM) Server.

### Intended Audience

This manual is intended for Interstage BPM Super Users, who are administrators managing Interstage BPM Server. It assumes that the reader has a working knowledge of the following:

- Operating system administration
- Database administration

### This Manual Contains

Here is a list of what is in this manual:

Chapter	Title	Description
1	Architecture Overview	Introduction to Interstage BPM, and description of the system components and their interaction.
2	Stopping and Starting Interstage BPM Server	Instructions for starting and stopping the Interstage BPM Server.
3	Configuring Interstage BPM Server	Description of how to change the configuration of the Interstage BPM Server.
4	System Configuration Parameters of the Interstage BPM Server	Description of the server parameters.
5	Properties of Interstage BPM Server Tenants	Description of the tenant-specific parameters.
6	Making a Backup of Your Data	Description of the data, which are to be stored.
Appendix A	Database Schema	Description of the database schema of Interstage BPM.
Appendix B	Message List of Interstage BPM Server	Description of the messages outputting to Status.log.
Appendix C	Troubleshooting	Description of the Interstage BPM log files and specific error situations and troubleshooting information.
	Glossary	Glossary of terms.

### Typographical Conventions

The following conventions are used throughout this manual:

---

Example	Meaning
command	Text, which you are required to type at a command line, is identified by <code>Courier</code> font.
<b>screen text</b>	Text, which is visible in the user interface, is <b>bold</b> .
<i>Reference</i>	Reference material is in <i>italics</i> .
Parameter	A command parameter is identified by <code>Courier</code> font.

## Other References

The following references for Interstage Business Process Manager are also available:

- *Release Notes*  
Contains an overview of Interstage Business Process Manager, setup tips, and late-breaking information that could not make it into the manuals.
- *Interstage Business Process Manager Server and Console Installation Guide*  
Describes software and hardware requirements, setup procedure for Interstage Business Process Manager Server and Console
- *Interstage Business Process Manager Developer's Guide*  
Describes how to use the Interstage Business Process Manager API to customize and extend Interstage BPM to fit the unique needs of your organization.
- *Interstage Business Process Manager Studio User's Guide*  
Explains how to model processes using the Interstage Business Process Manager Studio.
- *Interstage Business Process Manager Tenant Management Console Online Help*  
Explains how to use the Interstage Business Process Manager Tenant Management Console user interface.
- *Interstage Business Process Manager Console Online Help*  
Explains how to use the Interstage Business Process Manager Console user interface.
- *API Javadoc*  
This HTML documentation provides the API and syntax of the packages, interfaces and classes for developing custom applications or embedding Interstage Business Process Manager into other products.

## Abbreviations

The products described in this manual are abbreviated as follows:

- "Oracle WebLogic Server" is abbreviated as "WebLogic".
- "Interstage Business Process Manager" is abbreviated as "Interstage BPM".
- "Microsoft® Windows Server® 2008" and "Microsoft® Windows Server® 2012" are abbreviated as "Windows Server".
- "Oracle Solaris" might be described as "Solaris", "Solaris Operating System", and "Solaris OS" in this document.
- "Microsoft® SQL Server 2008 Standard Edition", "Microsoft® SQL Server 2008 Enterprise Edition", "Microsoft® SQL Server 2012 Standard Edition", "Microsoft® SQL Server 2012 Business

Intelligence Edition", and "Microsoft® SQL Server 2012 Enterprise Edition" are abbreviated as "Microsoft® SQL Server", "Microsoft SQL Server", or "SQL Server".

- "Oracle Database 11g R1 Standard Edition", "Oracle Database 11g R1 Standard Edition One", "Oracle Database 11g R1 Enterprise Edition", "Oracle Database 11g R2 Standard Edition", "Oracle Database 11g R2 Standard Edition One", "Oracle Database 11g R2 Enterprise Edition", "Oracle Database 12c R1 Standard Edition", "Oracle Database 12c R1 Enterprise Edition" are abbreviated as "Oracle".
- "JBoss Enterprise Application Platform" is abbreviated as "JBoss EAP".
- "Java Development Kit" and "Java SE Development Kit" is abbreviated as "JDK".

### **Export Controls**

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

# 1 Architecture Overview

Interstage Business Process Manager (Interstage BPM) is a server-based workflow engine with APIs (Application Programming Interfaces) for workflow application development. It empowers developers or systems engineers to embed a workflow engine into their own products or systems which implement Interstage BPM.

Some key features of Interstage BPM include:

- API, which allows customized applications to communicate with the workflow engine or existing products to be workflow-enabled.
- Enterprise-wide, scalable infrastructure for handling processes of all types
- Organizable and filterable universal to-do list
- Central location for documents relevant to a process

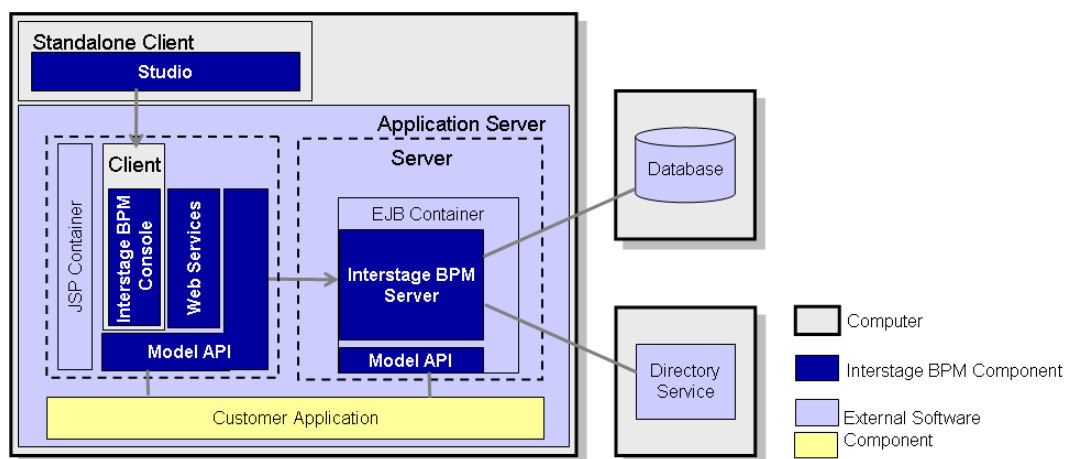
Interstage BPM can run on multiple application servers providing load balancing and failover capabilities for non-stop operation with nearly 100% reliability. One customer installation ran a production server non-stop, 7x24, and using careful 6-Sigma methods measured a server up-time to be 99.996% and the only reason it was not 100% was a small computer hardware maintenance issue. Therefore, Interstage BPM is ideally suited for large mission critical applications deployed on the leading J2EE-compliant application servers. Refer to section *System Architecture* on page 13 for more information.

Interstage BPM can be used together with the following integration components:

- **UDDI Repository:** UDDI provides storage infrastructure for Web Services registries, Meta model repositories, as well as data necessary for auditing, access security and versioning. Refer to section *UDDI Repository* on page 17 for more information.

## 1.1 Configuration Overview

Interstage BPM can be run in the following configuration:



**Figure 1: Configuration Overview**

The Interstage BPM Server operates with a Database and optionally a Directory Service. The Interstage BPM Clients are used to access the Interstage BPM Server.

You can setup the components that make up a complete Interstage BPM setup in various configurations:

- All systems are setup one and the same computer
- One or several of the following is setup on separate computers:
  - Interstage BPM Server and Console
  - Database
  - Directory Service
  - Studio

Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

**Note:** Model API must refer the NamingService of Interstage BPM Server. Therefore, if the customer application is referring its own NamingService that is available on another machine, it is necessary to integrate that NamingService with the NamingService of Interstage BPM Server. The NamingService of the customer application cannot independently refer the NamingService of Interstage BPM Server.

## 1.2 System Architecture

Interstage BPM basically consists of a Server and a Model API. Several connectivity options allow for the integration of third party tools and other systems. This section provides an overview of the Interstage BPM components and their interaction.

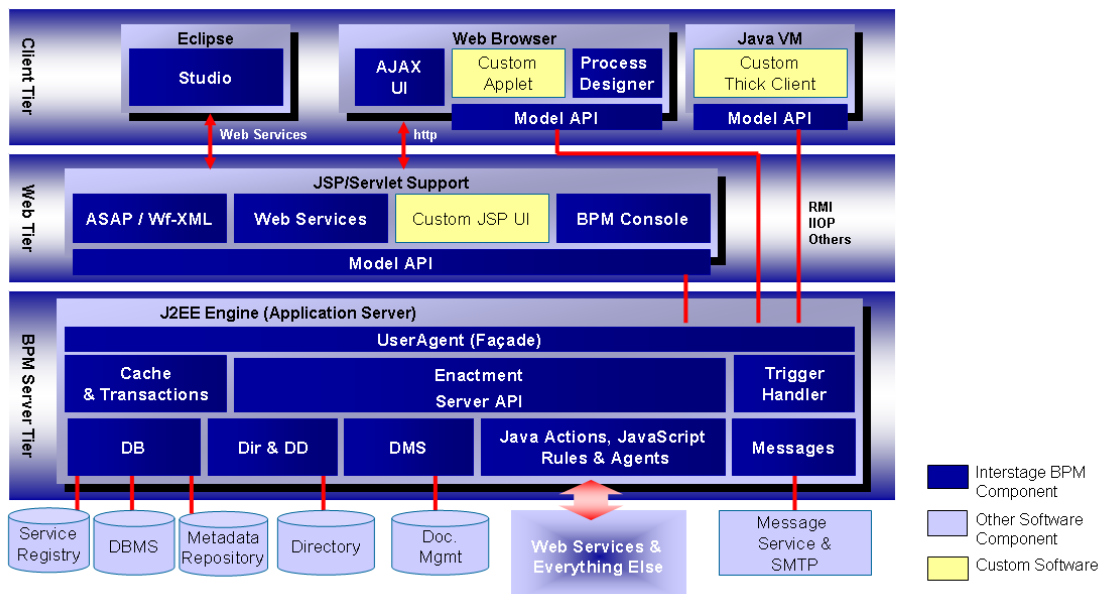


Figure 2: Architecture Overview

### 1.2.1 Interstage BPM Server Tier

The Interstage BPM Server is running inside an application server providing an Enterprise Java Bean (EJB) interface. The server negotiates interaction between users and other components, enacts

processes started by users, and notifies users of changes in status within a process. Interstage BPM can be configured using the standard capabilities of the application server, and managed using standard tools that work with application servers such as standard backup software, virus protection software, and distributed install software.

Interstage BPM is setup into a standard application server that isolates the application from operating system and database differences. Interstage BPM utilizes the facilities of the application server to provide, for example, for clustering, load balancing and failover capabilities. Standard application servers provide SNMP connectivity to allow monitoring of the server through SNMP. Interstage BPM can run on just about any operating system and in any hypervisor (e.g. VMware vSphere, Oracle VirtualBox, etc.) where the application server can be run. (The specific certified environments are listed in the release notes.)

The server is composed as a collection of EJBs that run in an application server, and make use of application server functionality. The Interstage BPM EJBs participate in container transactions so that the server and any client application can participate in the same transactions. Container-based transactions ensure a consistent state of the server. The only way to access the server is through the Model API, which allows for all forms of automated access: automated reporting, external system demand integration, and even automated testing of implemented processes.

The subsequent sections describe the EJBs in more detail.

### **User Agent (Façade)**

The User Agent EJB (UA bean) enables the client to “log in” to the Interstage BPM system and validates the client for further interaction with Interstage BPM. First, the client requests a UA bean to be created; next the server creates the UA bean and returns a handle to it to the client. Then the client provides user name, password and server name for login to the server; this information is validated either through Interstage BPM's local user management capabilities or using the functionality of a connected Directory Service.

A UserAgent instance represents the login session to the server. It holds information for that particular login session. As the client's agent, the UserAgent makes bean requests and method calls to the other Interstage BPM components on behalf of its client, i.e. it acts as a gateway for the model to access the process definition, process instance, work item, directory, and other objects. Therefore, the UA bean is also referred to as **Façade**, representing the interface between the server and the model.

Another function of the UA bean is to interpret the various filters on process definition, process instance and work item objects. Upon a client's option to log out, the UserAgent will do all the necessary cleanup of resources held on behalf of the associated client. In addition, the UA bean implements session synchronization.

The server requests a unique instance of the UA bean for each client that logs into Interstage BPM. It requests these instances from the EJB container that is part of the application server. In a manner of speaking, the Server bean is a factory that “produces” instances of UA beans.

### **Enactment Engine - Process Definition Interpreter**

The Process Definition Interpreter is the heart of the Interstage BPM Server. It is responsible for enacting a process defined with Interstage BPM. The server communicates with the Database adapter to maintain process state data, process instance and activity-related data, and process history information. The server controls database request queues.

There are two types of entity beans representing data objects holding the information about process definitions and instances. Upon enactment of a process, a Process Definition EJB is created, which

in turn creates Process Instance beans. Both beans implement the application server functionality of Bean-Managed-Persistence and Container Transactions.

They are not exposed to the model. All requests to the model pass through the UA bean(s).

Process data, e.g. information about the current state, is stored in and, on request, retrieved from the database. The server communicates with the database to maintain process state data, process and activity-relevant data.

## **Messaging**

A combination of message-driven beans (MDBs) and a Java class library implement the Interstage BPM's type system (Meta model). Process enactment events are encapsulated in JMS messages that the MDBs process. Interstage BPM makes use of the default application server functionality.

In Interstage BPM, message-driven beans (MDBs) realize the flow of information between the server components by means of asynchronous messages.

For example, there are the following MDBs:

- Enactment Message bean: When a process instance is created, this bean generates a message so that the client is informed about this.
- Email Dispatcher bean: Handles email messages to the client
- Action Agent bean: handles Action Agents

JMS connectivity and MDBs are the most effective way to integrate with standard Message Oriented Middleware (MOM) also known as an Enterprise Service Bus (ESB). Because messaging is a native part of Interstage BPM it is particularly easy and effective to integrate process through an ESB to other enterprise services.

## **Custom EJBs**

Any application can implement EJBs that run on the same application server setup as the Interstage BPM Server. Custom EJBs use the Model API, and Interstage BPM EJBs can call custom EJBs using Java Action.

## **User and Group Management**

Every user that is to work with Interstage BPM needs a user account and must be assigned to one or more groups. Groups are used to determine who is responsible for carrying out a task in a process.

Interstage BPM comes with its own user and group management capabilities. Interstage BPM also allows for connecting to a Directory Service. Depending on your choices when setting up the server, users are managed either in Interstage BPM's local user store or in a Directory Service. Groups can be managed in Interstage BPM's local group store, in a Directory Service or in both systems.

## **Persistence**

All process data and metadata is stored in a database, even when a process appears to the user to be running. The process instance can be in a running state for an unlimited amount of time. The number of processes instances that can be running at the same time is also nearly unlimited. The only limit to the number of processes that can run at once is the amount of disk space that the database needs to store the data. A relatively small process might take 100KBytes per process instance, and 10 million such process instances will take about 1TB of disk space. A more complicated process will take a proportional amount more, yet the real limitation on the number of processes remains only the space needed for the database.

The only time a process instance is read out of the database is when it needs to change state, for instance when a user completes a task, an event is received for that process instance, or a timer goes off causing an escalation or other timed activity. We refer to every such update as a transaction. The primary limit to scalability of a server is the limit to the number of transactions that can be performed in a unit time. The exact number of transactions per unit time depends upon the speed of the hardware, the speed of the IO, the latency of the network, the amount of data in each transaction, etc. As a rule of thumb, small servers can handle 10,000 to 20,000 transactions per hour, while large servers can reach 100,000 and possibly even 1 million transactions per hour. By counting the number of steps in a single process, you can estimate how many processes per hour you can process.

Persisting all process data in the database makes the servers stateless with regard to a given user. This means that there is no significant overhead caused by a user logging into the system. Adding users to the system is just a matter of adding the userids, and there is no critical dependency on the number of users. The system is limited instead by the number of transactions that it can handle. The exact number of transactions per unit time depends upon the speed of the hardware, the speed of the IO, the latency of the network, the amount of data in each transaction. For example, 100,000 users making one update per hour would produce virtually the same load as 100 users making 1000 updates per hour. The number of users, by itself, does not matter. Because the server does not hold any information outside of the database, a clustered environment can scale to any number of servers, and servers can be dynamically added and removed on demand.

The secret to the high reliability of the engine is the proper use of database transactions. All updates for a given operation are committed in a single quick database transaction. The process instance state is either completely updated, or not changed at all. This guarantees that the process instance is always in a consistent state, no matter what happens -- even unplugging the server in the middle of an update will not leave the process instance in an invalid state.

- **Database (DB)** adapter using the JDBC standard. The server provides the communication mechanism between the server and a database server. The database persistently stores and maintains all process information. The DB Adapter is responsible for the translation of server internal objects into persistent database formats. Included with Interstage BPM is an adapter that persists the structures in a relational database using JDBC and some stored procedures. If a non-relational database, or other exotic database, is required it would be possible to write a custom DB adapter to meet this need.

## Connectivity

The Interstage BPM architecture allows for the integration with third-party products. The server can communicate with the other components via "adapter classes". An adapter behaves as a converter that allows the server to speak to a common interface. Interstage BPM allows for connecting to the following:

- **Directory Service (Dir & DD)** adapter which implements an Interstage BPM specific interface to expand a user group into a list of individuals. The enactment engine uses this at runtime to determine who to give work items to. The Directory Adapter uses the LDAP standard. Currently, Microsoft® Active Directory and Oracle Directory Server Enterprise Edition is supported.  
The DD Framework Adapter is used by the User Agent at login time to authenticate users.
- **DMS** adapter which is used to interface the Interstage BPM system to external file systems using standard copy and transfer protocols. Forms, attachments, process definitions, etc. can be stored in a file system. A locator for such documents is stored in the attachments attributes of a process instance. An adapter for the most common protocol, WebDAV, is included in Interstage BPM by default and can be used to access most document management system. Access to other document management systems may require a custom DMS Adapter to be installed. For example,



installing a CMIS adapter would allow access to document management systems that support CMIS protocol.

- **Messages** using the SMTP standard. Email can be sent from the server to the SMTP mail server as a response to Interstage BPM events.
- **External Systems:**
  - **Java Actions and JavaScript:** You can implement Java Actions for connecting to any external system, such as CRM or ERP systems. Java Actions are extensions to the workflow engine. Java Actions are data structures in the process definition that tell the Interstage BPM Server how to call a particular Java method during execution. Java Actions make application integration easier and calls to external applications and adapters faster.
  - **Agents:** Agents in Interstage BPM are set up to run automatically and act asynchronously on your behalf. You can use Agents to access external systems such as legacy systems or Web Services, both inside and outside of company firewalls. Using Agents, you can incorporate these external services into your Interstage BPM process instances. This mode of integrating Java is particularly convenient when multiple retries may be required.
  - **Rules:** The Rules Engine Bridge is a Java Action that is included in Interstage BPM in order to invoke rules engines like the iLog JRules Engine. Rules have all the same capabilities that are available from JavaScript, so you can think of the rules as a kind of scripting engine.

## 1.2.2 Interstage BPM Web and Client Tier

The **Model API** is an abstraction over the server and provides a single unified API to the server. The Model API runs in the client process, and handles all the communications to the server.

Interstage BPM comes with several client applications, for example, the Studio and the BPM Console. Except for the Studio, the clients run in a servlet engine and are accessed using a Web browser. They are comprised of a combination of Java User Interface classes. Such client components are structured in two layers: a model layer (using the Model API) and a user interface layer (using Java User Interface classes). The model layer encapsulates the state of the client objects and interacts with the server. With using the Model API, you can develop your own clients and user applications. The Studio is a standalone process design tool that can be installed separately. The Studio is independent of application server functionality. It interacts with the Interstage BPM Server through the Web tier.

**Web Service** capabilities is included in the web tier.

Interstage BPM supports a kind of Web Services Interface that is known as an Asynchronous Web Services Interface. This is an implementation of a standard way to access process instances and other long running programs. The standard is known as the Asynchronous Service Access Protocol (ASAP).

Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details on the Interstage BPM Web Services.

## 1.3 UDDI Repository

A UDDI repository provides storage infrastructure for Web Services registries, Meta model repositories, as well as for data necessary for auditing, access security and versioning. In addition it provides a web-based interface to visualize reports that analyze the usage of Web Services in process instances, orchestrations and information integration queries. As a result, business analysts, architects and developers can all collaborate, eliminate business risk related to change in IT assets and avoid the disruption of critical business processes.

From a functional point of view a UDDI repository manages metadata generated from integration software, Web Service descriptions, application specific data, and in general it serves as a central store for documents in native XML and non-XML formats.

UDDI is an industrial standard and serves the known registry functionality such as publicizing, discovering and staging consumption of Web Services. Publishing, discovering and retrieving Web Services capabilities provided by Interstage BPM is based on standard UDDI interfaces, and therefore you can use CentraSite from Interstage BPM as a UDDI registry implementation.

WebDAV is another industrial standard and can be used for storing and retrieving development artefacts, which are stored in standard formats such as XPDL. Interstage BPM provides the capability of publishing metadata into WebDAV and therefore you can use CentraSite from Interstage BPM as a WebDAV repository implementation.

## 1.4 Interstage BPM as a Service (SaaS)

Interstage Business Process Manager offers the option of being used in SaaS (Software as a Service) mode. If you use Interstage BPM in SaaS mode, you can create multiple tenants and lease out Interstage BPM to these tenant organizations, who will use it as a service. Note the following:

- When Interstage BPM is set up, it is automatically used in the SaaS mode.
- An organization that leases out Interstage BPM to other organizations for use as a service is called a service provider.
- An organization that uses Interstage BPM as a service from the service provider will use Interstage BPM as a 'tenant'.
- A service provider user who administrates tenants is called a Super User. Functionality of a Super User is limited to only managing tenants through the Interstage BPM Tenant Management Console, and managing the Interstage BPM Server.
  - Interstage BPM Tenant Management Console setup is automatically performed when you setup Interstage BPM Console.
  - Information about using the Interstage BPM Tenant Management Console is included in the Interstage BPM Tenant Management Console Online Help
  - Information about managing the Interstage BPM Server is included in the *Interstage BPM Administration Guide*
- The Super User role cannot use or administrate Interstage BPM workflows.
- Default tenant is automatically setup when you setup Interstage BPM. Setting up a default tenant also sets up the default `System` application.
- **Even if you use Interstage BPM in the non-SaaS mode:**
  - The role of this Super User will be limited to managing Interstage BPM Server.
  - You will use all Interstage BPM functionality as a default tenant. You will not be allowed to create more than one tenant.
- Irrespective of whether you use Interstage BPM in SaaS mode or not, any operation on a workflow element will always be in the specific context of an application. For example, before you create a process definition or process instance, you need to choose an application. However, choosing an application is optional. To know about the behavior when an application is not chosen, please refer the `ApplicationSecurityMode` parameter in the *Interstage BPM Administration Guide*.

## 1.5 Interstage BPM Server Self-Healing

The Interstage BPM Server and tenants have self-healing capabilities that enable the system to diagnose the problem in the event of system initialization failures and automatically initialize the system when the error is resolved.

**Note:** If any error occurs due to invalid configuration of system properties, server startup will fail.

### Self-Healing during System Initialization

In case of error during system initialization due to unavailability of any external system component, the self-healing behavior for system-level and tenant-level will be as mentioned below:

- **System-level:** If any error occurs in the system initialization due to unavailability of any external system component such as Database, the Interstage BPM server automatically retries the initialization procedure with 30 second interval until it is recovered successfully. Please note that the system along with all tenants will be unavailable and user will not be able to perform any operations during this recovery period.
- **Tenant-level:** If a tenant fails to initialize due to unavailability of any external system component such as Directory Server, it retries the initialization procedure with 30 second interval until recovered successfully. During this period, only this tenant will not be operational but other tenants will remain unaffected and continue their regular operations. Also, the Tenant Management console is available.

On successful initialization, respective health checks will be performed for system as well as for each tenant at regular time intervals, as specified by the value of `HealthCheckInterval` system property. If any error is detected in this health check, the system or tenant becomes unavailable for operations and starts self-healing process to retry the initialization. For details, refer *HealthCheckInterval* on page 48.

## 2 Stopping and Starting Interstage BPM Server

This chapter describes how to start and stop the Interstage BPM Server. The procedure for starting and stopping the server depends on the application server that you are using.

### 2.1 Stopping and Starting Interstage BPM for WebLogic

The following sections describe how to stop and start the Interstage BPM Server when using WebLogic Application Server.

#### 2.1.1 Stopping Interstage BPM for WebLogic

**Prerequisite:** When you stop the Interstage BPM Server it automatically logs out from the database and the Directory Server. Make sure that all Interstage BPM Clients are logged out from the Interstage BPM Server before stopping it.

The procedure for stopping Interstage BPM depends on whether you have a single server setup or a cluster setup.

- **To stop a single server:**

- a) Stop the Interstage BPM Server by running the shutdown script.

Operating System	Script
Windows	stopManagedWebLogic.cmd
UNIX	stopManagedWebLogic.sh

- b) If you need to stop the WebLogic Administration server, run the shutdown script provided by the application server:

Operating System	Script
Windows	stopWebLogic.cmd
UNIX	stopWebLogic.sh

All scripts above are located in the `<MW_HOME>/user_projects/domains/<Domain Name>/bin` directory.

- **To stop a cluster:**

- a) Stop the Interstage BPM Server by running the shutdown script located in `<engine directory>/server/deployment/WLS-Cluster:`

Operating System	Script
Windows	StopiFlowServer_Managed.bat <WebLogic Administrator Name> <WebLogic Administrator Password>
UNIX	StopiFlowServer_Managed.sh <WebLogic Administrator Name> <WebLogic Administrator Password>

For `<WebLogic Administrator Name>` `<WebLogic Administrator Password>`, specify the user account of the administrator for the WebLogic Administration Server.

- b) If you need to stop the WebLogic Administration server, run the shutdown script located in `<engine directory>/server/deployment/WLS-Cluster:`

Operating System	Script
Windows	stopWebLogic.bat
UNIX	stopWebLogic.sh

Alternatively, you can run the shutdown script provided by the application server in `<MW_HOME>/user_projects/domains/<Domain Name>/bin.`

## 2.1.2 Starting Interstage BPM for WebLogic

**Prerequisite:** Make sure, that the database server and the Directory Server are running before you start the Interstage BPM Server. If the database and/or Directory Server are on remote computers, start the appropriate services on each. Also, since by default Interstage BPM is setup in SaaS mode, ensure you have switched on Java Security. For details, refer the 'Switching On Java Security' topic in the *Interstage BPM Server and Console Installation Guide*.

The procedure for starting Interstage BPM depends on whether you have a single server setup or a cluster setup.

- **To start a single server:**

- a) Start the WebLogic Administration Server by running the startup script located in `<MW_HOME>/user_projects/domains/<Domain Name>/bin` directory.

Operating System	Script
Windows	startWebLogic.cmd
UNIX	startWebLogic.sh

On Windows with WebLogic V12.1.1 or earlier:

- You can start the server by selecting **Start > Programs > Oracle WebLogic > User Projects > <Domain Name> > Start Admin Server for WebLogic Server Domain.**

Upon successful initialization of the WebLogic Administration Server, the following message appears near the end of the WebLogic Console Window:

```
<Server started in RUNNING mode>
```

- b) Start the Interstage BPM Server by running the startup script located in `<MW_HOME>/user_projects/domains/<Domain Name>/bin` directory with the user who has access rights to `<engine directory>`(i.e. User who ran the setup script).

Operating System	Script
Windows	run<Server Name>.cmd Example: runIBPMServer.cmd
UNIX	run<Server Name>.sh Example: runIBPMServer.sh

For `<Server Name>`, specify the name that you defined when setting up the Interstage BPM Server.

- c) When asked by the startup script, enter the name and password of the administrator for the WebLogic Administration Server.

Upon successful initialization of the WebLogic Server, the following message appears near the end of the Console Window:

```
<Server started in RUNNING mode>
```

- **To start a cluster:**

- a) Start the WebLogic Administration Server by running the startup script located in `<engine directory>/server/deployment/WLS-Cluster:`

Operating System	Script
Windows	startWebLogic.bat
UNIX	startWebLogic.sh

Alternatively, you can run the startup script provided by the application server in

```
<MW_HOME>/user_projects/domains/<Domain Name>/bin.
```

Upon successful initialization of the WebLogic Administration Server, the following message appears near the end of the WebLogic Console Window:

```
<Server started in RUNNING mode>
```

- b) Start the Interstage BPM Server by running the startup script located in `<engine directory>/server/deployment/WLS-Cluster` with the user who has access rights to `<engine directory>` (i.e. User who ran the setup script).

Operating System	Script
Windows	iFlowServer_Managed.bat <WebLogic Administrator Name> <WebLogic Administrator Password>
UNIX	iFlowServer_Managed.sh <WebLogic Administrator Name> <WebLogic Administrator Password>

This ensures that required environment variables are set before actually starting the server.

For `<WebLogic Administrator Name>` `<WebLogic Administrator Password>`, specify the user account of the administrator for the WebLogic Administration Server.

You can start the cluster servers in any order.

- c) To examine whether the Interstage BPM Server is running check the `Trace.log` file located at `<engine directory>/server/instance/default/logs`

If the server starts successfully, you will see the following message at the end of the file:

```
<Date> ... Interstage BPM Server ... is ready.
```

If the server is not running, refer to section *Interstage BPM Server Fails to Start* on page 116.

## 2.2 Stopping and Starting Interstage BPM for JBoss

The following sections describe how to stop and start the Interstage BPM Server when using JBoss Application Server.

## 2.2.1 Stopping Interstage BPM for JBoss

### Prerequisite:

Make sure that all Interstage BPM Clients are logged out from the Interstage BPM Server before stopping it.

You stop the Interstage BPM Server by stopping the JBoss Application Server.

### To stop the application server, do one of the following:

- Display the Command Prompt window in which the JBoss Application Server is running. Press <Ctrl>-C and wait for the server to stop.

**Note:** Do not stop the JBoss Application Server by closing the Command Prompt window in which the server is running.  
Some components may not stop properly and you may not be able to start the JBoss Application Server again.

## 2.2.2 Starting Interstage BPM for JBoss

**Prerequisite:** Make sure, that the database server and the Directory Server are running before you start the Interstage BPM Server. If the database and/or Directory Server are on remote computers, start the appropriate services on each. Also, since by default Interstage BPM is setup in SaaS mode, ensure you have switched on Java Security. For details, refer the 'Switching On Java Security' topic in the *Interstage BPM Server and Console Installation Guide*.

You start the Interstage BPM Server by starting the JBoss Application Server.

### To start the application server, do the following:

- Run the following script located in the <JBoss Installation Directory>/bin directory:

Operating System	Script
Windows	domain.bat
UNIX	domain.sh

For more details, refer to the *JBoss Application Server Documentation*.

To examine whether the Interstage BPM Server is running check the `Trace.log` file located at <engine directory>/server/instance/default/logs

If the server starts successfully, you will see the following message near the end of the file:

```
<Date> ... Interstage BPM Server ... is ready.
```

If the server is not running, refer to section *Interstage BPM Server Fails to Start* on page 116.

---

## 3 Configuring Interstage BPM Server

This chapter provides instructions for changing the configuration of Interstage BPM Server.

The configuration parameters of the Interstage BPM Server define its operating environment. They define the server's operation modes, map the locations of error logs, command scripts, and configuration files and store connection information for external systems like a Directory Service or a Metadata Repository.

The server's configuration parameters are stored in the Interstage BPM database. Interstage BPM comes with configuration export and import scripts. The Tenant Management Console allows you to change parameter values. Refer *Changing Parameter Values Using Tenant Management Console* on page 24 for more information. When you, for the first time, add a parameter to the configuration which is not set to the Interstage BPM Server, it is necessary to use configuration export and import script. Refer *Using the Configuration Export and Import Scripts* on page 25 for information about using configuration export and import scripts.

**Note:** You can use the Interstage BPM Tenant Management Console to change these parameters. For instructions on accessing the Tenant Management Console, refer the *Interstage BPM Server and Console Installation Guide*.

### 3.1 Configuring the Email Listener

You must configure the Email Listener to use Interstage BPM Email Integration. Use these instructions to configure your Email Listener if you want the ability to complete tasks using your Email Client (Microsoft Outlook for example).

To configure your Email Listener, update the following Interstage BPM Properties according to the instructions found in the following sections of this chapter and the descriptions for these parameters that can be found in the chapter of this guide entitled *Properties of Interstage BPM Server Tenants*:

- EmailListenerAutoReplyEnabled
- EmailListenerDeleteInvalidMessages
- EmailListenerEmailAddress
- EmailListenerEnabled
- EmailListenerPassword
- EmailListenerPollingInterval
- EmailListenerPOPPort
- EmailListenerPropertiesFile
- EmailListenerServerHost
- EmailListenerUserName
- EmailStyleSheetFile
- ServerEmailAddress

### 3.2 Changing Parameter Values Using Tenant Management Console

**Prerequisite:** The Interstage BPM Console and the Interstage BPM Server are running.

Tenant Management Console allows you to change System parameter values while the server is running. Any changes made using the Tenant Management Console will take effect immediately.



**Note:** You can change both, System parameter values and Tenant parameter values using the Tenant Management Console.

- For information about System parameters, refer *System Configuration Parameters of the Interstage BPM Server* on page 44.
- For information about Tenant parameters, refer *Properties of Interstage BPM Server Tenants* on page 59.

**Note:** If you do not have Interstage BPM Console setup done on your local system, use the export and import configuration scripts to edit the parameter values. Refer *Using the Configuration Export and Import Scripts* on page 25 for more information about using export and import configuration scripts.

**Note:** Before you make any changes to the Server configuration, export a copy of the server configuration file as a back-up. In case you make errors while editing parameters, Interstage BPM Server may not be able to start or errors may occur at runtime. In such a case you can import the back-up file which contains a working configuration. For information about using export and import configuration scripts, refer *Using the Configuration Export and Import Scripts* on page 25.

**To change System parameter values using the Tenant Management Console:**

1. Access the Tenant Management Console using the following URL:

`http://<hostname>:<port>/console/TenantManager.page`

On JBoss and WebLogic Application Servers, use port number 49950.

2. Log in as a Super User.
3. Click the **System Properties** button.  
**System Properties** pop-up is displayed.
4. Edit the required parameter values.

**Note:** Parameter values are case sensitive.  
Be careful when changing parameter values. If you specify invalid values, the Interstage BPM Server may not be able to start or errors may occur at runtime.

5. Click **Save**.
6. In a clustered or load-balanced system, each server node must be configured individually. Therefore, repeat steps 1-5 for the other server nodes.

### 3.3 Using the Configuration Export and Import Scripts

**Prerequisite:** The database server is running.

Interstage BPM comes with configuration export and import scripts that allow you to change the server configuration. You can use these scripts to add parameters that are not present in the configuration or remove parameters from the configuration. If the server fails to start or denies login due to configuration errors, you can use the scripts to import a working configuration into the database. One convenient use of this capability is to take a snapshot of the configuration at a point in time to save as a backup and for future reference.

You can add or edit both System as well as Tenant properties using the export and import scripts.

The Interstage BPM Server is not required to be running for the configuration export and import scripts to be used.

**To use the configuration export and import scripts:**

1. Export the server configuration by running the export script located in `<engine directory>/server/deployment/bin:`

Operating System	Configuration Export Script
Windows	<ul style="list-style-type: none"> <li>• For System Properties:  <code>exportProperties.bat &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt;</code>  <b>Example:</b>  <code>exportProperties.bat MyServerConfiguration.properties SomeUserName SomePassword</code> </li> <li>• For Tenant Properties:  <code>exportProperties.bat &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt; &lt;TenantName&gt;</code>  <b>Example:</b>  <code>exportProperties.bat MyServerConfiguration.properties SomeUserName SomePassword Default</code>  <b>Note that &lt;TenantName&gt; is case-sensitive</b> </li> </ul>
UNIX	<ul style="list-style-type: none"> <li>• For System Properties:  <code>exportProperties.sh &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt;</code>  <b>Example:</b>  <code>exportProperties.sh MyServerConfiguration.properties SomeUserName SomePassword</code> </li> <li>• For Tenant Properties:  <code>exportProperties.sh &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt; &lt;TenantName&gt;</code>  <b>Example:</b>  <code>exportProperties.sh MyServerConfiguration.properties SomeUserName SomePassword Default</code>  <b>Note that &lt;TenantName&gt; is case-sensitive</b> </li> </ul>

For `<File Name>`, specify the file name and, optionally, the path of the configuration file to be generated. If you do not specify a path, the configuration file is written to the current directory.

For `<Database User Name>` and `<Password>`, specify the database user account that you defined during setup of the Interstage BPM Server.

**Note:** Before you make any changes to the configuration file, create a copy of the configuration file you exported as a back-up. In case you make errors while editing parameters and import the file, Interstage BPM Server may not be able to start or errors may occur at runtime. In such a case you can use the back-up file which contains a working configuration.

2. Make the required changes in the generated configuration file.

If you want to add parameters, use the following syntax:

```
<Parameter Name>=<Parameter Value>
```

<p><b>Note:</b></p> <ul style="list-style-type: none"><li>• Parameter names and values are case sensitive.</li><li>• Any backslashes "\" or colons ":" used in parameter values must be escaped by backslashes. For example, a server address is specified like this: <pre>ibpmhost\:49950</pre></li><li>• Some configuration parameters have a <code>&lt;Hostname&gt;</code> or <code>&lt;Hostname&gt;.&lt;Servername&gt;</code> suffix appended to their names. These suffixes are automatically removed during the export and are automatically appended during the import. Do not add these suffixes to the parameter names.</li></ul>
--

Passwords in the generated configuration file are encrypted for security reasons. If you change a password, you are recommended to encrypt it before importing the configuration file. Passwords are not automatically encrypted during the import. For details on how to encrypt passwords, refer to the *Interstage Business Process Manager Server and Console Installation Guide*.

3. Import the updated configuration file into the database by running the import script located in

```
<engine directory>/server/deployment/bin:
```

Operating System	Configuration Import Script
Windows	<ul style="list-style-type: none"> <li>• For System Properties:  <pre>importProperties.bat &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.bat MyServerConfiguration.properties SomeUserName SomePassword</pre> </li> <li>• For Tenant Properties:  <pre>importProperties.bat &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt; &lt;TenantName&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.bat MyServerConfiguration.properties SomeUserName SomePassword Default</pre> </li> </ul>
UNIX	<ul style="list-style-type: none"> <li>• For System Properties:  <pre>importProperties.sh &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.sh MyServerConfiguration.properties SomeUserName SomePassword</pre> </li> <li>• For Tenant Properties:  <pre>importProperties.sh &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt; &lt;TenantName&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.sh MyServerConfiguration.properties SomeUserName SomePassword Default</pre> </li> </ul>

For `<File Name>`, specify the file name and, if required, the path of the updated configuration file. You need to specify the path if the configuration file is not located in the current directory.

For `<Database User Name>` and `<Password>`, specify the database user account that you defined during setup of the Interstage BPM Server.

4. In a clustered or load-balanced system, each server node must be configured individually. Therefore, repeat steps 1-3 for the other server nodes.
5. Restart the Interstage BPM Server for the changes to take effect.  
For details, refer to chapter *Stopping and Starting Interstage BPM Server* on page 20.

### 3.4 Changing the Database and Directory Service Configuration

**To make changes in your database or Directory Service configuration:**

1. Export the server configuration so that you have a record of any customizations that you have made. To do so, run the export script located in `<engine directory>/server/deployment/bin:`

Operating System	Configuration Export Script
Windows	<ul style="list-style-type: none"> <li>• For System Properties:  <pre>importProperties.bat &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.bat MyServerConfiguration.properties SomeUserName SomePassword</pre> </li> <li>• For Tenant Properties:  <pre>importProperties.bat &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt; &lt;TenantName&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.bat MyServerConfiguration.properties SomeUserName SomePassword Default</pre> <p><b>Note that &lt;TenantName&gt; is case-sensitive</b></p> </li> </ul>
UNIX	<ul style="list-style-type: none"> <li>• For System Properties:  <pre>importProperties.sh &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.sh MyServerConfiguration.properties SomeUserName SomePassword</pre> </li> <li>• For Tenant Properties:  <pre>importProperties.sh &lt;File Name&gt; &lt;Database User Name&gt; &lt;Password&gt; &lt;TenantName&gt;</pre> <p><b>Example:</b></p> <pre>importProperties.sh MyServerConfiguration.properties SomeUserName SomePassword Default</pre> <p><b>Note that &lt;TenantName&gt; is case-sensitive</b></p> </li> </ul>

For <File Name>, specify the file name and, optionally, the path of the configuration file to be generated. If you do not specify a path, the configuration file is written to the current directory.

For <Database User Name> and <Password>, specify the database user account that you defined during setup of the Interstage BPM Server.

2. Un-setup the Interstage BPM Server from the application server using the Interstage BPM un-setup script.  
Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.
3. Re-setup the Interstage BPM Server using the new database or Directory Service configuration information.  
Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

4. If required, reconfigure the Interstage BPM Server as explained in section *Changing Parameter Values Using Tenant Management Console* on page 24 and section *Using the Configuration Export and Import Scripts* on page 25.

## 3.5 Configuring the Document Management System (DMS)

The Document Management System (DMS) is used to store Workflow Applications, attachments to process instances and optionally other documents.

When you setup the Interstage BPM Server, you specify one or more DMS root directories. You can add additional DMS root directories by modifying the `DmsCollections.xml` configuration file.

### To add a DMS root directory:

1. Create a new directory in your DMS.
2. Edit the `DmsCollections.xml` configuration file. By default, the file is located in `<engine directory>/server/instance/default/tenants/<tenant name>/resources`.
3. Add a new `<Dms>` section to the file:

```
<Dms>
  <Path> ... </Path>
  <ImplementationClass> ... </ImplementationClass>
  <Name> ... </Name>
</Dms>
```

This is the information that needs to be specified in the section's elements:

- `<Path>`: This is the **DMS Root** directory. Specify the full path of the directory in your DMS, for example the full path of a file system directory or the URL of a WebDAV directory.

**Note:** If you specify a file system directory, ensure this directory is specific to a tenant, and resides inside the `<engine directory>/server/instance/default/tenants/<tenant name>` directory. Also note that the `<DMSRoot>` directory will contain all your workflow application projects under the `apps` folder.

**Note:**

- Make sure that you specify the path with a slash at the end to ensure proper handling of attachments in the Interstage BPM Console and Studio.
- If you are running Interstage BPM Server on Windows and you wish to specify a remote directory, use one of the following formats:
 

```
//<Remote Computer Name>/<Directory Name>/
\\<Remote Computer Name>\<Directory Name>\
```

Do not specify a network drive for example like in `P:\SomeRemoteDirectory`. Network drives cannot be accessed by the Interstage BPM Server.

- `<ImplementationClass>`: Specify the appropriate DMS adapter class. For a file system directory, specify `com.fujitsu.iflow.dmsadapter.impl.FileSystemDmsSessionImpl`. For a WebDAV directory, specify `com.fujitsu.iflow.dmsadapter.impl.WebDAVDmsSessionImpl`. If you are using a custom DMS adapter, specify the class name of that adapter.

**Note:** To use WebDAV directory as DMS, ensure that at least one File DMS directory (local path folder) is present as the first DMS path in `DmsCollections.xml`.

- `<Name>`: Define a unique symbolic name for the directory.

The symbolic name is optional, but highly recommended. If you define a symbolic name, the Interstage BPM Server uses that name instead of the directory's physical path. This is useful for sharing data between different Interstage BPM Servers and for clustered or load-balanced systems.

Say you want to set up a test server and a production server. Each server has its own DMS root directory, and the directories have different physical paths. If you are using the same symbolic name on both servers, you can transfer XPD files having external files attached to them between the servers without having to adjust the physical DMS root path.

In a clustered or load-balanced system, the DMS root directories are located on a central file server where they are accessed by all server nodes. The server nodes are allowed to map those directories to different physical paths as long as they use the same symbolic name for a directory. For details on how to set up the DMS for a clustered or load-balanced system, refer to the *Interstage Business Process Manager Server and Console Installation Guide*.

- `<UserId>`: You can specify user ID to access DMS, if you are using WebDAV directory.
- `<UserPwd>`: You can specify password of the user name used to access DMS, if you are using WebDAV directory. You can specify either the plain text password or encrypted password. However, it is recommended that you specify encrypted password. You can use the `TextEncrypter` utility to encrypt the value. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

**Note:** The `<UserId>` and `<UserPwd>` are optional properties. If you do not specify, Interstage BPM Server login user ID and password will be used.

- `<AuthType>`: You can specify the type of authentication required for WebDAV. Valid values are `Basic` and `NTLM`. This property is optional, if you do not specify then `Basic` authentication will be used.
- `<UserDomain>`: Specify the domain name of the repository user with which user should be authenticated. This property is mandatory, if you are using NTLM authentication for WebDAV directory.

#### 4. Restart the Interstage BPM Server.

For details, refer to chapter *Stopping and Starting Interstage BPM Server* on page 20.

This is an example of an entry for a file system directory on Windows:

```
<DmsCollections>
  <Dms>
    ...
  </Dms>
  <Dms>
    <Path>c:\Fujitsu\InterstageBPM\server\instance\default\tenants\MyTenant\AnotherDMSRoot</Path>

    <ImplementationClass>
      com.fujitsu.iflow.dmsadapter.impl.FileSystemDmsSessionImpl
    </ImplementationClass>
    <Name>AnotherDMSRoot</Name>
  </Dms>
</DmsCollections>
```

This is an example of an entry for a file system directory on UNIX:

```
<DmsCollections>
  <Dms>
    ...
```

```

</Dms>
<Dms>
  <Path>/opt/FJSVibpm/instance/default/tenants/MyTenant/AnotherDMSRoot</Path>

  <ImplementationClass>
    com.fujitsu.iflow.dmsadapter.impl.FileSystemDmsSessionImpl
  </ImplementationClass>
  <Name>AnotherDMSRoot</Name>
</Dms>
</DmsCollections>

```

This is an example of an entry for a WebDAV directory that requires Basic authentication:

```

<DmsCollections>
  <Dms>
    ...
  </Dms>
  <Dms>
    <Path>http://www.example.com/MyDMSRoot</Path>
    <ImplementationClass>com.fujitsu.iflow.dmsadapter.impl.WebDAVDmsSessionImpl

    </ImplementationClass>
    <Name>AnotherDMSRoot</Name>
    <UserId>User Name</UserId>
    <UserPwd>User Password</UserPwd>
  </Dms>
</DmsCollections>

```

This is an example of an entry for a WebDAV directory that requires NTLM authentication:

```

<DmsCollections>
  <Dms>
    ...
  </Dms>
  <Dms>
    <Path>http://www.example.com/MyDMSRoot</Path>
    <ImplementationClass>com.fujitsu.iflow.dmsadapter.impl.WebDAVDmsSessionImpl

    </ImplementationClass>
    <Name>AnotherDMSRoot</Name>
    <AuthType>NTLM</AuthType>
    <UserId>User Name</UserId>
    <UserPwd>User Password</UserPwd>
    <UserDomain>User Domain</UserDomain>
  </Dms>
</DmsCollections>

```

**Note:** If you use Sharepoint Server as WebDAV DMS repository, you can not use following special characters in a folder or file name:

~, #, %, &, \*, { }, \, :, <, >, ?, /, |, "

Refer to the *Microsoft Support site for SharePoint Server documentation* for details.

**Note:** If you are using the secure protocol (https), then you need to configure the WebDAVKeyStore property. For details, refer to section *WebDAVKeyStore* on page 74.

## 3.6 Integrating Interstage BPM with Other Servers

You can integrate Interstage BPM Server with other servers. You can then start subprocesses running on one of the servers from parent processes running on another. The integrated servers communicate



with each other using ASAP (Asynchronous Service Access Protocol) or SWAP (Simple Workflow Access Protocol).

**Note:** You are recommended to use ASAP when the processes run on integrated Interstage BPM Servers with other servers.

If you are integrating only remote Interstage BPM servers and are running them in a secure environment, integration involves configuring the Interstage BPM Server user.

**To integrate remote Interstage BPM Servers that are running in a secure environment:**

1. Make sure that all the Interstage BPM Servers that you want to integrate are using the same Interstage BPM Server User Name and Password.

For more information, refer to section *ServerUserName* on page 70.

2. Add the following parameters to the configuration of all the Interstage BPM Servers that you want to integrate:

```
SWAPLinkageUserName=ibpm_server1
```

```
SWAPLinkagePassword=<Password of ibpm_server1>
```

For instructions on how to add parameters, refer to section *Using the Configuration Export and Import Scripts* on page 25.

You can use the `TextEncrypter` utility to encrypt the password. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

3. Restart all the Interstage BPM Servers that you want to integrate.

You can now start subprocesses running on one of the Interstage BPM Servers from parent processes running on another. For instructions on creating the process definitions for both the parent and child processes in Interstage BPM, refer to the *Interstage Business Process Manager Studio User's Guide*.

### 3.6.1 Integration of Interstage BPM to Interstage BPM

The integration of Interstage BPM to Interstage BPM is interactive and involves the following communications:

- Interstage BPM to Interstage BPM CREATEPROCESSINSTANCE
- Interstage BPM to Interstage BPM (Process) COMPLETE

#### Interstage BPM to Interstage BPM CREATEPROCESSINSTANCE

A parent process in Interstage BPM starts a subprocess on a remote Interstage BPM Server. Interstage BPM will be able to start the remote subprocess only if the Interstage BPM Linkage User

(`ibpm_server1`) is authenticated on the remote Interstage BPM server as a normal Interstage BPM user.

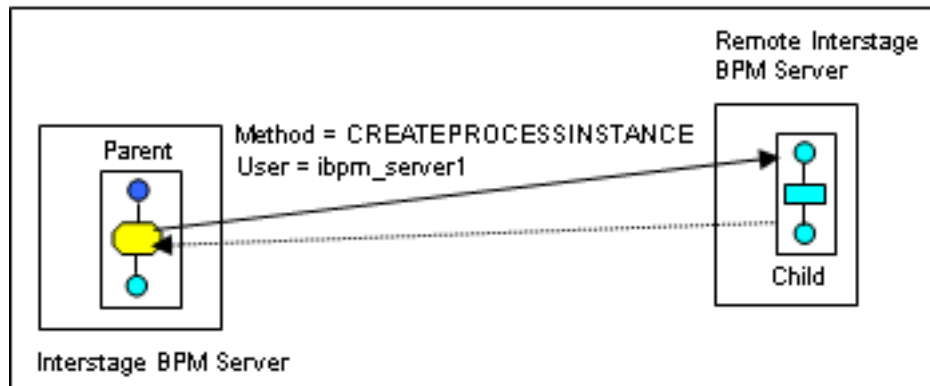


Figure 3: Interstage BPM to Interstage BPM CREATEPROCESSINSTANCE

### Interstage BPM to Interstage BPM (Process) COMPLETE

The Interstage BPM remote subprocess notifies its parent process in Interstage BPM when it has completed. This notification can be received only if the Interstage BPM Linkage User (`ibpm_server1`) is authenticated on the local Interstage BPM server as a normal Interstage BPM user.

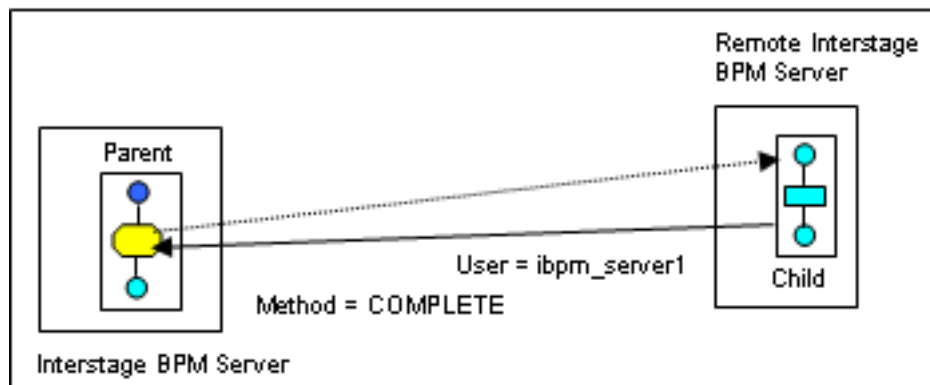


Figure 4: Interstage BPM to Interstage BPM (Process) COMPLETE

## 3.6.2 Setting a Different Interstage BPM Linkage User

A different user can be used for the Interstage BPM Linkage User.

**To set a different Interstage BPM Linkage User:**

1. Add the following parameters to the configuration of the Interstage BPM Server:

- `SWAPLinkageUserName=<user name>`
- `SWAPLinkagePassword=<password>`

Set these parameters equal to the user name and password that you want to use as the Interstage BPM Linkage User:

- On the local Interstage BPM Server, specify a user of the remote server that can create subprocesses.
- On the remote server, specify a user of the local Interstage BPM Server.

For instructions on how to add parameters, refer to section *Using the Configuration Export and Import Scripts* on page 25.

You can use the `TextEncrypter` utility to encrypt the password. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

2. Make sure that this user is authenticated in Interstage BPM to perform all of the necessary operations.

## 3.7 Configuring Business Calendars

Interstage BPM allows you to create timers that trigger certain events associated with the timer upon its expiration. The Business Calendars feature allows you to create business timers. A business timer is a special type of timer that will only “count” business hours and days and expires only during business hours. Business hours and days are specified in a Business Calendar.

You can create and use your own custom business calendars or modify the default business calendar to meet your needs.

You may create as many Business Calendars as necessary to meet the needs of your organization and use a different Business Calendar for every process definition or process instance.

The following sections explain the format of business calendars. Refer to the *Interstage Business Process Manager Studio User's Guide* and *Interstage Business Process Manager Developer's Guide* for information on how to use business calendars.

### 3.7.1 Business Calendar Format

Business Calendars are like properties or `.ini` files. They are composed of name-value pairs on a line of the file terminated by a semicolon (;). These name-value pairs specify the business days and hours. To indicate ranges, e.g. start and end times, a comma (,) is used.

The name of the calendar corresponds to the name of the calendar file. The calendar file is given a `.cal` file extension, so it can be recognized as a Business Calendar. Business Calendars are application-specific.

When an application is created, a default calendar (named `Default.cal`) is created in the `<DMSRoot>/apps/<application id>` directory. If no other calendar is configured for use with Interstage BPM, you can still add a business timer to your Interstage BPM process instances, and the default calendar will specify the business days and hours. The default calendar also provides an example of a fully functional Business Calendar.

**Note:** After changing calendar files, restart the Interstage BPM Server.

You can use several parameters to specify business hours:

- `EVERYDAY`  
Specifies the default business hours.
- `<Day of the Week>`  
Usually used to remove days from the work week, e.g. Saturday and Sunday.

- `<Date>`

Usually used to remove holidays from the work year.

These parameters have the following priority:

`<Date>` > `<Day of the Week>` > `EVERYDAY`

**Example:** You can set up your business hours as follows:

```
EVERYDAY=9:00,17:00;
```

```
FRI=9:00,14:00;
```

```
2007/06/08=09:30,13:00;
```

In this example, the effective business hours for Friday, June 08 are from 09:30 to 13:00. For all other Fridays, business hours are from 09:00 to 14:00. For all other days, business hours are from 09:00 to 17:00.

The following sections provide a detailed description of all parameters used in the calendar file.

## EVERYDAY

This parameter is required. It defines the default business hours for every day of the week.

**Example:** `EVERYDAY=9:00,17:00;`

## <Day of the Week>

This parameter defines the default business hours for a particular day of the week. This will override the `EVERYDAY` setting for that day of the week.

**Example:** You can specify the business hours for a particular day (i.e. `SUN`, `MON`, `TUE`, `WED`, `THU`, `FRI`, `SAT`) as follows:

```
FRI=9:00,16:00;
```

No hours specified means that that day of the week is not a working day.

This parameter is usually used to remove Saturday and Sunday from the work week. This is accomplished by setting the day of the week to a null value. In the following example Saturday and Sunday are removed from the work week:

```
SAT=; SUN=;
```

## <Date>

This parameter defines the business hours for a specific date. This overrides any settings for `EVERYDAY` or for `<Day of the Week>`.

**Format:** `yyyy/mm/dd` (year/month/day)

This parameter is usually used to remove holidays from the work year.

**Example:** `2007/01/01=;`

It is also used to specify special hours like a long lunch hour for a special event.

**Example:** `2007/12/01=9:00,12:00;15:30,17:00;`

## CALENDAR\_BEGIN

This parameter is required. It defines the first day the calendar is valid. It sets the calendar boundary in that the calendar cannot be used before the date specified by this parameter. If timer expiry is set before this time, then the process will go into error state with an array out of bounds error.

Default value: `CALENDAR_BEGIN=2003/01/01;`

**Note:** The calendar duration can be a maximum of 10 years. Therefore the start time (`CALENDAR_BEGIN`) and the end time (`CALENDAR_END`) of the business calendar may lie maximally 10 years apart.

## CALENDAR\_END

This parameter is required. It defines the last day the calendar is valid. It sets the calendar boundary in that the calendar cannot be used past the date specified by this parameter. If timer expiry is set beyond this time, then the process will go into error state with an array out of bounds error.

Default value: `CALENDAR_END=2010/12/31;`

**Note:** Either of the following methods can be used to set the Daylight Saving Time or the Summer Time, and the Time Zone.

- Set `TIMEZONEID` (Recommended)
- Set `DST` and `TIMEZONE`

If you specify both, an exception will be thrown.

The `DST` and `TIMEZONE` parameters support the timezone and DST settings used in previous versions of the Interstage BPM to provide backward compatibility. If you use these parameters, you need to manually modify the calendar for every year. Thus, it is recommended to use the `TIMEZONEID` parameter to specify the timezone and adjust the hours for Daylight Saving Time or Summer Time.

## TIMEZONEID

This parameter is used to define the time zone ID that is to be used to interpret the current time of day in the calendar for that time zone, relative to Greenwich Mean Time (GMT). The time zone includes region specific offsets due to Daylight Saving Time or Summer Time. Thus, when you specify 8:00 AM in this calendar, it is interpreted as 8 AM in the specified time zone at that time of year.

If you specify the time zone ID for using Daylight Saving Time or Summer Time, the corresponding time of year is automatically used.

This parameter is optional. If it is not specified, then the value of `TIMEZONE` parameter is used.

If this parameter is left blank (i.e. `TIMEZONEID=;`), then time zone ID of the machine on which Interstage BPM Server is running, will be used. Following are the examples of timezone IDs:

- `TIMEZONEID=America/New_York;`
- `TIMEZONEID=America/Los_Angeles;`
- `TIMEZONEID=America/Indiana/Indianapolis;`
- `TIMEZONEID=America/Chicago;`
- `TIMEZONEID=America/Panama;`
- `TIMEZONEID=America/St_Vincent;`
- `TIMEZONEID=America/Toronto;`
- `TIMEZONEID=Australia/Melbourne;`
- `TIMEZONEID=Australia/Victoria;`
- `TIMEZONEID=Australia/Sydney;`
- `TIMEZONEID=Australia/Canberra;`

- `TIMEZONEID=Asia/Kolkata;`
- `TIMEZONEID=Asia/Tokyo;`
- `TIMEZONEID=Europe/Amsterdam;`
- `TIMEZONEID=Europe/Paris;`
- `TIMEZONEID=Europe/Rome;`
- `TIMEZONEID=Indian/Maldives;`
- `TIMEZONEID=Japan;`
- `TIMEZONEID=US/Arizona;`
- `TIMEZONEID=US/Pacific;`
- `TIMEZONEID=US/Eastern;`

The valid values of `TIMEZONEID` parameter are based on Java `TimeZone` class. Use the `TimeZone.getAvailableIDs()` API to get the list of available timezone IDs, as shown below.

```
import java.util.TimeZone;
public class MainClass
{
    public static void main(String[] a)
    {
        String[] ids = TimeZone.getAvailableIDs();
        for (String id : ids)
        {
            System.out.println(id);
        }
    }
}
```

## DST

This parameter is used to adjust the hours for Daylight Saving Time or Summer Time.

Format: `<Date>=DST(<Time offset>)`

For `<Time offset>`, you can specify a value from 0 to 4 hours.

Example: To adjust the hours forward on 20th April and back on 19th October, you specify the following entries:

```
2006/04/20=DST(1);
```

```
2006/10/19=DST(0);
```

Restrictions:

- You cannot combine `DST(<Time offset>)` with hour specifications. The following calendar entry, for example, is not valid:  
`2006/10/20=DST(1);9:00,12:00;13:00,17:30;`
- You can specify DST setting for single year only. The business calendar can contain only one `DST(non-zero)` and one `DST(zero)` entries in the sequence as specified in the above example.
- The same DST setting is used for every year. If you will use different value next year, you need to manually modify the value of DST parameter.

## TIMEZONE

This parameter defines the time zone that is to be used to indicate the current time of day in the calendar relative to Greenwich Mean Time (GMT).

Example: Pacific Time in the U.S. is GMT - 8 hours, so the Pacific Timezone is specified as  
`TIMEZONE=-8:00;`

This parameter is optional. If it is not specified or left blank (i.e. `TIMEZONE=;`) then time zone of the machine on which Interstage BPM Server is running, will be used.

### 3.7.2 Shift Example

The Business Calendar allows for swing and graveyard shifts that start before midnight and end after midnight.

When you set up a calendar, say you need to give a default time for Mondays. The shift may go from 8:00 PM to 4:00 AM. The entire shift is considered a Monday shift, even though it crosses midnight and occurs half in Tuesday. This shift is designated as being from 20:00 to 28:00. The corresponding calendar entry looks like this:

```
MON=20:00,28:00;
```

On the other hand, if this shift starting Monday night is supposed to be associated with Tuesday, then you use -4:00 to 4:00. In this case, a Monday holiday will have effect on the shift that starts Sunday night, but will have no effect on the shift starting Monday night. The corresponding calendar entry looks like this:

```
TUE=-4:00,4:00;
```

When specifying shifts, note the following:

- You can specify shift ranges (that is, the start and end hour of a shift) from -24 to +48 hours.
- You cannot specify overlapping shifts. The following calendar entry, for example, is not valid:

```
FRI=10:00,17:30;10:30,18:00;
```

## 3.8 Configuring Log Files

### 3.8.1 System-level log files and Tenant-specific log files

#### Log Levels and Categories

All messages are tagged with log-levels. A log-level is the severity of a message. The following levels (severities), sorted from highest to lowest severity, are available for log messages:

- `FATAL` - Errors due to which Interstage BPM server cannot function are logged under this level.
- `ERROR` - Error messages during server operation are logged under this level.
- `WARN` - Errors due to user-operations or execution of application program are logged under this level.
- `INFO` - Audit trace information is logged under this level.
- `DEBUG` - Debugging messages for Interstage BPM support teams are logged under this level.

All messages are also tagged with categories. A category denotes the part of Interstage BPM to which a message relates. Currently, messages are grouped into the following categories:

- `AUDIT` - All messages related to user login-logout operations, Process Scheduler and Agents are grouped under `AUDIT` category.
- `SYSTEM` - Other messages are grouped under the `SYSTEM` category.

Though all messages are tagged with categories, not all log files display categories of error messages.

#### Log Files

The following log files are available in Interstage BPM:

- `Error.log` - this log file collects Interstage BPM system-level errors. System administrators can ascertain whether Interstage BPM Server is working correctly by monitoring this file. If the Interstage BPM Server is working correctly, no logs will be entered in this file. This log file exists separately for system-level and tenant-level.
- `Trace.log` - this file collects trace information of user operations (API usage) and external integrated systems such as java actions, java scripts, agents, and so on. This log file is useful for application developers, for example, when they need to view the exception trace when an error occurred due to invalid user operation. This log file exists separately for system-level and tenant-level.
- `Audit.log` - this file collects user-login information. System administrators can use this file to know which user is logged in and which user is logged out. It also collects messages related to Process Scheduler and Agents. This log file exists separately for system-level and tenant-level.
- `Support.log` - An Interstage BPM support-cache continually collects debug information (all error messages with log-level greater than or equal to `DEBUG`). On occurrence of an error with a log-level higher than specified in the `SupportLogThreshold` parameter of Interstage BPM, the cache flushes all its messages to `Support.log`.

The size of the support-cache can be set using the `SupportLogBufferSize` parameter. For information about these parameters, refer the *Interstage BPM Server Administration Guide*.

`Support.log` exists only for system-level. It is helpful for anybody wanting to debug (such as Interstage BPM support team members).

### Log File Locations

By default, these files are located at the following location on the Interstage BPM Server computer.

- System-level log files are located at `<engine directory>/server/instance/default/logs`
- Tenant-specific log files are located at `<engine directory>/server/instance/default/tenants/<tenant name>/logs`

### Customizing Log Files

Some log file parameters (such as file size, location, content) can be customized using the following Interstage BPM configuration parameters.

- `ServerLogsDirectory` (System-level and tenant-level)
- `ErrorLogRecycleSize` (System-level and tenant-level)
- `ErrorLogLevel` (System-level and tenant-level)
- `TraceLogRecycleSize` (System-level and tenant-level)
- `TraceLogLevel` (System-level and tenant-level)
- `AuditLogRecycleSize` (System-level and tenant-level)
- `SupportLogRecycleSize` (System-level)
- `SupportLogThreshold` (System-level)
- `SupportLogBufferSize` (System-level)
- `LogFileRecycleNumber` (System-level and tenant-level)

## 3.8.2 Status log files

The information about running status of Interstage BPM Server is recorded to `Status.log`. This log output is isolated from the settings of system-level log files and tenant-specific log files.



However, you cannot customize the log file location, the maximum size or the number of backup files, so that the behavior of status log files is described below:

#### Log File Location

```
<engine directory>/server/instance/default/logs
```

The maximum size of the status log file is 1 MByte. When the file size reaches to maximum size, Status.log is renamed to Status.log.1, and then the latest information is outputted to Status.log continuously. After this action is repeated 3 times, Status.log.3 is deleted, and then Status.log.2 is renamed to Status.log.3 repeatedly.

About the message of status log, the format is fixed to the followings, therefore you can monitor running status of Interstage BPM Server by using monitoring software to check the message of this file.

```
[<Date>] <Label>: <Error Level>: <Message ID>: <Message Text>
```

The details of each item are shown below.

Name	Details
Date	The date and time when server status changes.
Label	Fixed value mentioned below: FSP_INTS-BPM_IBPM-SERVER
Error Level	A message type. A system administrator can decide what type of problem occurs according to the message type. The following message types are outputted. ERROR, WARNING, INFO
Message ID	A unique identifier of message. A system administrator can specify what problem occurs according to this value. Refer to <i>Message List of Interstage BPM Server</i> on page 109.
Message Text	The details of message.

#### Example of message output

```
[09/01/2015 12:00:00] FSP_INTS-BPM_IBPM-SERVER: ERROR: 00011: Failed to initialize tenant: TENANT=Default
```

## 3.9 Dynamically Reloadable Properties

If an API call (such as `WFSuperSession.setTenantPropeties()`, or `WFAdminSession.updateIflowPropeties()`) changes an Interstage BPM property, for the changed value to reflect to the Interstage BPM Server:

- The Interstage BPM Server usually needs to be re-started
- In case of a tenant-specific property, that tenant needs to be activated

However, there are certain properties that do not require the steps mentioned above for the Interstage BPM Server to reflect their changed values; if an API call changes their values, the changed values are dynamically reloaded in the Interstage BPM memory.

## 3.10 Changing Tenant Resource Permissions

**This information is applicable for SaaS mode only.**

In SaaS mode, if switching on Java security, the following permissions are set to system resources by the custom class loader of Interstage BPM. If necessary, an administrator can change user-defined permissions.

Default permissions switching on Java security:

- Some permission targets of `RuntimePermission` and `SerializablePermission`
- "read", "write", "delete", "execute" actions to the following files and directories:
  - Files and sub-directories in directories specified in `DmsCollections.xml` file that is the configuration file of Document Management System (DMS).
  - `<engine directory>/server/instance/default/tenants/<tenant name>/attachments` directory
- "read" action to the following files and directories:
  - Files and sub-directories in `<engine directory>/server/instance/default/tenants/<tenant name>/classes` directory
  - The `.jar` and `.zip` files in `<engine directory>/server/instance/default/tenants/<tenant name>/lib/ext` directory
  - Files and sub-directories in `<engine directory>/server/instance/default/classes` directory
  - The `.jar` and `.zip` files in `<engine directory>/server/instance/default/lib/ext` directory
- User-defined permissions:
  - The permissions defined in `<engine directory>/server/instance/default/resources/policy.xml` file.

As an administrator, you can edit the `policy.xml` file to change permissions.

Edit `policy.xml` file in `<engine directory>/server/instance/default/resources/` directory. This file is provided by Interstage BPM itself, and described about permissions by XML format. The format of the `policy.xml` file should be as follows:

```
<permissions>
  <grant>
    <permission
      className="permission class name"
      target="target name"
      action="action name"/>
    ...
  </grant>
</permissions>
```

The description of each attribute is as follows:

For details of values of attributes, refer to the security documents of JDK that you use.

---

Attribute	Description
className	Specify class name of the permission
target	Specify target name of the permission which is related to the class name
action	Specify action of the permission which is related to the class name

If granting not enough permissions, an exception is outputted to server log file at runtime of Interstage BPM like the following:

```
java.security.AccessControlException: access denied (java.io.FilePermission /tmp read)
```

This means that `java.io.FilePermission` class is lacking for "read" action to /tmp directory. In this case, add the lacking permission to `policy.xml` file as follows:

```
<permissions>
  <grant>
    ...
    <permission
      className="java.io.FilePermission"
      target="/tmp"
      action="read"/>
    </grant>
  </permissions>
```

**Note:** Permissions in `policy.xml` affect all tenants.

## 4 System Configuration Parameters of the Interstage BPM Server

The configuration parameters of the Interstage BPM Server define its operating environment. They define the server's operation modes, map the locations of error logs, command scripts, and configuration files. Those configuration properties that define the entire Interstage BPM System are called System Properties. This chapter covers only the System Properties. Every tenant using Interstage BPM has properties that are unique to that tenant. Those properties are covered in the next chapter.

Many of the server's configuration parameters are similar to parameters used in a query. Server information, for example, specifies the attributes of the Interstage BPM Server. This information is the same throughout this setup of Interstage BPM and will be used by Interstage BPM for authentication among other Server-related tasks. By providing this framework for creating queries, the configuration provides the means by which many different functions can be implemented.

The server's configuration parameters are stored in the Interstage BPM database. Some of the parameters were set when you carried out the initial system configuration using the Interstage BPM setup script.

If you need to reconfigure the server or change parameters that cannot be configured in the setup script, you can change the server configuration manually. For instructions, refer to section *Changing Parameter Values Using Tenant Management Console* on page 24 and section *Using the Configuration Export and Import Scripts* on page 25.

**Note:** Parameter names and values are case sensitive.  
Be careful with any configuration changes. If you specify incorrect values, the Interstage BPM Server or other components may not be able to start or errors may occur at runtime.

This appendix lists the names and values of parameters that you may change or add. In addition, some parameters are listed that you must not change but that you may need to know about. Parameters that are used for internal purposes only are not listed. Many parameters have initial values provided by the server. These initial values, if any, are listed at the beginning of each parameter description below. You can override initial values by editing the parameter's value.

### 4.1 AllowNonDMSFileAttachmentsInEmailAction

Initial value: `false`

Setting this property to `true` allows the Email action to attach a file irrespective of its location on the server. If set to `false`, the Email action can attach files only in the DMS directories.

### 4.2 AnalyticIntegrationSuspendHours

Initial value: `0`

This parameter is used to specify the amount of time that can lapse before Analytics Integration is suspended. If an event message is not collected by Analytics Sensor. Analytics Integration will resume after the message is collected. For this parameter, the default value is `0`, the minimum value is `1` hour, and the maximum value `1000000` hours.

When an outside range value is specified, the event message is always written in DB.

Prepare the DB capacity in which the event message that occurred for the period, specified for this parameter, can be stored.

## 4.3 ApplicationSecurityMode

Initial value for SaaS mode: `Secure`

Initial value for non-SaaS mode: `Relax`

**Note:** This parameter should not be removed and its value should not be blank.

Application security, that is, whether application data for one tenant is segregated from applications belong to other tenants or not, is controlled by the `ApplicationSecurityMode` parameter.

- If set to `Secure` application data for each tenant is segregated from other tenants. Multiple tenants can be created.
- If set to `Relax`, only a single, default tenant can be activated. You can not create or delete tenant.

To segregate an application from the other applications, you need to choose application. However, choosing an application is optional. Cross-application data access is permitted if no application is chosen.

If no application is chosen:

- While creating or copying a process definition,
  - In `Relax` mode, the default `System` application is selected.
  - In `Secure` mode, the default `System` application is selected. If `System` application does not exist, then an exception will be thrown.
- You can access and work with process definitions and instances from all applications.
- Objects from all applications are retrieved while processing lists and process trigger events.
- Objects from all applications are notified while processing a notification event.

## 4.4 AuditLogRecycleSize

Initial value: `1000`

This parameter specifies the maximum size of the `Audit.log` file in Kbytes. If the log file reaches the specified size, it will be closed and renamed by adding a time stamp. A new log file with the original log file name will be opened and logging will continue with the new log file. The renamed log files can then be removed or deleted as appropriate without stopping the Interstage BPM Server. The `LogFileRecycleNumber` parameter specifies the maximum number of times a log file can be recycled.

For example if you set `AuditLogRecycleSize` as follows:

```
AuditLogRecycleSize=1000
```

Then, any log file that reaches the size of 1000 KB (1 MB) will be closed and renamed by adding a time stamp.

If the parameter is not set, the default value 1000 is used.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.5 ClusterAutomateFailoverEnabled

Initial value: `true`

This property controls whether automated failover is enabled or disabled in a cluster.

---

If set to `true`, automated failover is enabled.

- The status (running/failed) of all nodes is checked at regular intervals, as specified by the value of the `ClusterHeartbeatInterval` property. For details, refer *ClusterHeartbeatInterval* on page 46.
- If a node fails, unhandled events from that failed node are handled by other active nodes. The number of unhandled events to be handled at a time is specified by the value of the `ClusterInterceptEvents` property. For details, refer *ClusterInterceptEvents* on page 46.

**Note:** If set to `false`, automated failover is disabled; no status-check of nodes takes place, and if a node fails, unhandled events from that failed node are not handled by other active nodes.

This is a dynamically reloadable property, that is, you do not need to re-start the Interstage BPM server when you change this property.

## 4.6 ClusterHeartbeatInterval

Initial value: 30

Minimum value: 10

This property specifies the time interval (in seconds) between successive status checks of all nodes in a cluster. For example, if this property is set to 30, the status (running/failed) of each node is checked every 30 seconds.

If an invalid value (less than 10) is specified, the default value of 30 seconds is used.

**Note:** The status of nodes is checked only if the `ClusterAutomateFailoverEnabled` property is enabled. For details, refer *ClusterAutomateFailoverEnabled* on page 45.

This is a dynamically reloadable property, that is, you do not need to re-start the Interstage BPM server when you change this property.

## 4.7 ClusterInterceptEvents

Initial value: 30

Minimum value: 10

This specifies the number of unhandled events of a failed node that will be handled at a time by other active nodes during automated failover. Only positive integers greater than or equal to 10 are allowed.

If an invalid value (less than 10) is specified, the default value of 30 is used.

**Note:** The automated failover will take place only if the `ClusterAutomateFailoverEnabled` property is set to `true`. For details, refer *ClusterAutomateFailoverEnabled* on page 45.

This is a dynamically reloadable property, that is, you do not need to re-start the Interstage BPM server when you change this property.

## 4.8 ClusterMulticastPort

Initial value:

This parameter specifies the port number used by the multicast group to which the Interstage BPM Servers belong. A multicast group is used to distribute information among the Interstage BPM Servers in a clustered or load-balanced system.

The parameter is used in a clustered or load-balanced system only. **If clustering, ensure you add a value for this parameter.** Use the default value of 5000, or another value of your choice.

## 4.9 ClusterMulticastTopic

Initial value:

This parameter specifies the multicast group address, which is used to identify the members of a multicast group. A multicast group is used to distribute information among the Interstage BPM Servers in a clustered or load-balanced system.

The multicast group address is an IP address in the range 224.0.0.0 to 239.255.255.255, inclusive. Each member of the group uses the same multicast group address. The address 224.0.0.0 is reserved and must not be used.

The parameter is used in a clustered or load-balanced system only. **If clustering, ensure you add a value for this parameter.** Use the default value of 225.4.5.6, or another value of your choice.

## 4.10 DBSQLQueryTimeOut

Initial value: 300

This parameter specifies the maximum time in seconds which should be permitted for a database JDBC call.

Ensure that the value of `DBSQLQueryTimeOut` is always lower than the application server's transaction timeout value.

## 4.11 EncryptionKey

Initial value:

This parameter is not configured by default. It is required when you specify a special encryption key using `EncryptPassword.bat` or `EncryptPassword.sh`. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details on `EncryptPassword.bat` or `EncryptPassword.sh`.

## 4.12 ErrorLogLevel

Initial value: `ALL:ERROR`

This parameter specifies the default log level and log category for `Error.log`. Errors having severities equal to and above the specified log level will be logged in the file.

This parameter has the format `<LogCategory>:<LogLevel>`. You can also specify multiple levels for different categories using comma-separated values, such as `<LogCategory>:<LogLevel>,<LogCategory>:<LogLevel>`.

The value `ALL:WARN,SYSTEM:INFO` specifies that the following errors will be logged to `Trace.log`

- Errors of all categories, with log level `WARN` or above (that is, `WARN`, `ERROR`, `FATAL`)
- Errors of `SYSTEM` category, with log level `INFO` or above (that is, `INFO`, `WARN`, `ERROR`, `FATAL`)

(Note that `ALL` is not a pure category, but can be used as a keyword to refer to all existing categories.)

**Note:** Configuration for other category will override the configuration for `ALL` category.

For more details about log levels and categories, refer *Configuring Log Files* on page 39.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.13 ErrorLogRecycleSize

Initial value: 1000

This parameter specifies the maximum size of the `Error.log` file in Kbytes. If the log file reaches the specified size, it will be closed and renamed by adding a time stamp. A new log file with the original log file name will be opened and logging will continue with the new log file. The renamed log files can then be removed or deleted as appropriate without stopping the Interstage BPM Server. The `LogFileRecycleNumber` parameter specifies the maximum number of times a log file can be recycled.

For example if you set `ErrorLogRecycleSize` as follows:

```
ErrorLogRecycleSize=1000
```

Then, any log file that reaches the size of 1000 KB (1 MB) will be closed and renamed by adding a time stamp.

If the parameter is not set, the default value 1000 is used.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.14 HealthCheckInterval

Initial value: 600

Minimum value: 60

This property specifies the time interval (in seconds) between successive health checks of the Interstage BPM Server and all the tenants while they are running. For example, if this property is set to 800, the health check is executed every 800 seconds at system-level as well as tenant-level.

If 0 (zero) is specified, then no health check is executed. If the parameter is not set, the default value 600 is used. If an invalid value (less than 0 or between a range of 1 - 59) is specified, an error will occur.

This is a dynamically reloadable property. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.15 JMSNamingProviderURL.<Hostname>

Initial value: <URL of an Interstage BPM Server>

**Note:** This parameter should not be removed and its value should not be blank.

This parameter specifies the URL of an Interstage BPM Server instance. The URL is required to receive messages sent to the JMS topics. Its value depends on the application server being used and on the parameters that you specified when setting up the Interstage BPM Server.

The name of the parameter is different for single-server setup and clustered or load-balanced systems. In a single-server setup, the parameter's name is `JMSNamingProviderURL.<Hostname>`. In a clustered or load-balanced system, the parameter's name is `JMSNamingProviderURL.<Hostname>.<Server Name>`.



## 4.16 JSExtCmdsClasses

Initial value:

This parameter includes a class that enables calls to Web Services through JavaScript.

To extend the set of JavaScript commands using Java classes, specify fully qualified class names of your extension classes. This can be a comma separated list. Ensure that the class is placed in the following directory:

```
<engine directory>/server/instance/default/classes
```

**Note:** • In a clustered or load-balanced system, you must place the customized class in the `classes` subdirectory of each server node.

## 4.17 JSExtCmdsFile.<Hostname>

Initial value: `<engine directory>/server/instance/default/resources/jscommands.txt`

This file extends the scripting commands within the Interstage BPM Script Interpreter and Web Services Script Interpreter. When the server is initially started, it reads this file and extends the Script Interpreter to include these commands. To further extend the scripting language, add additional commands to this file, or add additional files with new commands. If you add additional files, they must be added to this parameter in a comma-separated list.

These definitions could be stored anywhere. The Interstage BPM setup process puts these files into a subdirectory of the `<engine directory>` by default.

The name of the parameter is different for single-server setup and clustered or load-balanced systems. In a single-server setup, the parameter's name is `JSExtCmdsFile.<Hostname>`. In a clustered or load-balanced system, the parameter's name is `JSExtCmdsFile.<Hostname>.<Server Name>`.

**Note:** • This parameter also exists as specific to each application; an application developer can configure the application-level parameter within the application BAR file.

## 4.18 KPIListenerConfigFile.<Hostname>

Initial value: `<engine directory>/server/instance/default/resources/kpilistener.conf`

This parameter identifies the KPI Listener configuration file and its location. The KPI Listener configuration file provides the scan interval that defaults to one minute. After every minute, the listener will check the status of the KPI threshold information.

The name of the parameter is different for single-server setup and clustered or load-balanced systems. In a single-server setup, the parameter's name is `KPIListenerConfigFile.<Hostname>`. In a clustered or load-balanced system, the parameter's name is `KPIListenerConfigFile.<Hostname>.<Server Name>`.

## 4.19 LogFileRecycleNumber

Initial value: 3

This parameter specifies the maximum number of times a log file will be recycled.

- If the parameter is set to 0, there is no restriction on the number of times a log file can be recycled.
- If the parameter is set to 1, a log file is created only once. When the maximum size of the log file is reached, the log file is cleared and reused. No archive log files are created.

- If the parameter is set to a value greater than 1, archive log files are created and recycled. When the maximum number of log files is reached, the oldest archive file is automatically deleted and a new archive log file is created.

If the parameter is not set, the default value 3 is used.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.20 MaxProcessDefinitionCacheSize

Initial value: 20

This parameter specifies the maximum number of process definitions that can be stored in server-side cache.

While configuring this variable make sure that:

- The value of `ObjectCacheEnabled` parameter should be set to `true`.
- The minimum memory size required for caching process definitions is available on the JVM heap. This can be calculated as:

```
Required memory size for caching process definitions =  
MaxProcessDefinitionCacheSize * Size of one process definition
```

The default maximum size of a process definition is 0.5 MB. If you set

`MaxProcessDefinitionCacheSize=20`, then the minimum JVM heap required will be as:

`20 * 0.5 MB = 10 MB`

Thus, the minimum memory size required for caching process definitions that should be available on JVM heap is 10 MB. Otherwise it may lead to an out of memory exception.

If you set the value of this parameter to 0, no process definition will be store in server-side cache.

## 4.21 MaxProcessInstanceCacheSize

Initial value: 20

This parameter specifies the maximum number of process instances that can be stored in server-side cache.

While configuring this variable make sure that:

- The value of `ObjectCacheEnabled` parameter should be set to `true`.
- The minimum memory size required for caching process instances is available on the JVM heap. This can be calculated as:

```
Required memory size for caching process instances = MaxProcessInstanceCacheSize  
* Size of one process instance
```

The default maximum size of a process instance is 4 MB. If you set

`MaxProcessInstanceCacheSize=20`, then the minimum JVM heap required for this will be as:

`20 * 4 MB = 80 MB`

Thus, the minimum memory size required for caching process instances that should be available on JVM heap is 80 MB. Otherwise it may lead to an out of memory exception.

If you set the value of this parameter to 0, no process instance will be store in server-side cache.

## 4.22 ObjectCacheEnabled

Initial value: `true`

This parameter enables or disables storage of the internal structure of objects in a cache.

- If you want to enable the server-side caching, set its value to `true`.

Enabling storage in server-side cache improves the performance but also increase the memory consumption. If you want to reduce the amount of memory consumed, change the value of the `MaxProcessDefinitionCacheSize` parameter or `MaxProcessInstanceCacheSize` parameter. For details, refer to sections *MaxProcessDefinitionCacheSize* on page 50 and *MaxProcessInstanceCacheSize* on page 50 respectively.

- If you want to disable the server-side caching, set its value to `false`.

## 4.23 ProactiveNotificationEnabled

Initial value: `true`

When the value of this parameter is set to `true`, the model-side notifications regarding changes in process definitions, process instances, and work items are enabled. If you want to disable model-side notification then you need to set this parameter to `false`. After disabling notifications, Interstage BPM Server does not post any notifications to the to the Model API. Thus, none of the clients receives any notifications. As a consequence the `WFObjectListListener` and `ProcessInstanceListener` APIs cannot be used.

## 4.24 ProactiveSQNotificationEnabled

Initial value: `true`

When the value of this parameter is `true`, the model-side notifications regarding change in sequence ids of process definitions and process instances are enabled. If value for this parameter is set to `false`, the sequence id notifications are disabled. This can result in old data while using model APIs, since the Model cache is not notified of any changes to process definitions and process instances in the server.

<b>Note:</b> Refer section <i>Model-side Notifications Used by Model API</i> of <i>Interstage Business Process Manager Developer's Guide</i> for information about Model cache.
---

## 4.25 SearchIndexUpdateInterval

Initial value: `60`

This parameter specifies the time interval (in minutes) between successive synchronization of the search index (used in full text search) with the latest data in the Interstage BPM database. The updated search index is used in full text search of the process instance matching the text string that user have specified as the search criteria.

If 0 (zero) is specified, then search index will never be updated and hence full text search feature will not be enabled.

If the parameter is not set, the default value 60 is used.

If an invalid value (less than 0) is specified, an error will occur.

## 4.26 ServerBaseURL

Initial value: `http://<hostname>:<port>/console/`

This parameter specifies the URL of the Interstage BPM Console. Its value is set during the initial Console configuration, which is explained in the *Interstage Business Process Manager Server and Console Installation Guide*.

**Note:** In this URL, the final '/' is a required character.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.27 ServerEmailBaseURL

Initial value: `http://<hostname>:<port>/console/`

When Interstage BPM uses email to notify a user about a new work item, the task is presented in the email message as a hyperlink to the server. The hyperlink is generated by prepending the value of this parameter to that of the `ServerEmailClientPage` parameter. The combined URL points to the email client. The parameter's value is set during the initial configuration of the Interstage BPM Console, which is explained in the *Interstage Business Process Manager Server and Console Installation Guide*.

If you develop a custom email client, specify the appropriate value for this parameter, for example:

```
ServerEmailBaseURL.<Hostname>=http://<hostname>:81/console/
```

**Note:** In this URL, the final '/' is a required character.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.28 ServerGNEmailCustomizerClass

This parameter specifies the location and name of an installation-supplied class to allow the Interstage BPM Server to create customized email messages for work item notification, rather than the default Interstage BPM sample note. This class should implement the `com.fujitsu.iflow.server.customize.intf.GNEmailCustomizerInterface` interface.

**Example:** `ServerGNEmailCustomizerClass=MyCustomizedEmail`

If the custom class is in a package, example:

```
ServerGNEmailCustomizerClass=mypackage.email.MyCustomizedEmail
```

Place the customized class in the following directory:

```
<engine directory>/server/instance/default/classes
```

**Note:** • In a clustered or load-balanced system, you must place the customized class in the `classes` subdirectory of each server node.

You can configure email notifications at application level. For details, refer section *Using Email Notifications* in *Interstage BPM Developer's Guide*.

## 4.29 ServerHost.<Hostname>

Initial value: <hostname>

**Note:** This parameter should not be removed and its value should not be blank.

This parameter identifies the host name (machine name) where Interstage BPM is setup. This value is set when you setup the Interstage BPM Server.

The name of the parameter is different for single-server setup and clustered or load-balanced systems. In a single-server setup, the parameter's name is `ServerHost.<Hostname>`. In a clustered or load-balanced system, the parameter's name is `ServerHost.<Hostname>.<Server Name>`.

## 4.30 ServerLogsDirectory

Initial value: logs

**Note:** This parameter should not be blank.

This parameter identifies the location where all server log files will be generated.

For system-level, the value of this parameter is relative to <SERVERROOT>, that is, <engine directory>/server/instance/default/

For tenant-level, the value of this parameter is relative to <TENANTHOME>, that is, <engine directory>/server/instance/default/tenants/<tenant name>/

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.31 ServerMaxUnhandledEvents

Initial value: 10000

Allowed values: Integers greater than 0

This parameter sets the maximum number of allowed unhandled events in the Interstage BPM Server. If the Interstage BPM Server reaches the maximum threshold set by this parameter, certain user operation will be denied to stop accumulation of more events. Calling those operations will throw an Exception until the Interstage BPM Server reaches below the threshold.

The operations denied are the creating and starting of process instances and the `makeChoice()` option of work items. Action agents configured with automatic `makeChoice()` options will also be hold and will keep retry until the Interstage BPM Server reaches below the threshold. The `RetryInterval` will be configured in the action agent.

**Note:** If you specify an invalid value, the Interstage BPM Server uses the initial value 10000.

## 4.32 ServerName

Initial value:

This is the symbolic name of a specific Interstage BPM Server instance.

In a single-server setup, you may change this value to any name that suits your needs. In a clustered or load-balanced system, refer to the *Interstage Business Process Manager Installation Guide* for information on how to set this parameter. The parameter `ServerName` is case sensitive.

If the parameter is not set, the default value `<Hostname>` is used.

### 4.33 ServerRoot.<Hostname>

Initial value: `<engine directory>`

**Note:** This parameter should not be removed and its value should not be blank.

This is the location where you have copied the engine directory, for example `C:/Fujitsu/InterstageBPM` on Windows and `/opt/FJSVibpm` on UNIX.

The name of the parameter is different for single-server setup and clustered or load-balanced systems. In a single-server setup, the parameter's name is `ServerRoot.<Hostname>`. In a clustered or load-balanced system, the parameter's name is `ServerRoot.<Hostname>.<Server Name>`.

### 4.34 ServerSharedRoot.<Hostname>

Initial value: `<engine directory>`

**Note:** This parameter should not be removed and its value should not be blank.

This is the shared ResourceRoot location of load balanced nodes.

In a single-server setup, the name of this parameter is `ServerSharedRoot.<Hostname>`.

In a clustered or load-balanced system, the name of this parameter is `ServerSharedRoot.<Hostname>.<Servername>`.

### 4.35 SMTPPassword

Initial value:

This is the password of the user name used for SMTP authentication.

You can use the `TextEncrypter` utility to encrypt the value. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

### 4.36 SMTPServerHost

Initial value: `<smtp_hostname>`

This parameter identifies the host name (machine name) where the mail server is located. The Interstage BPM Server uses this name to locate the mail server.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

### 4.37 SMTPServerPort

Initial value: 25

This is the port number that the mail server listens on. If your mail server uses a different port number, you need to change the value accordingly.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.38 SMTPUserName

Initial value:

This value identifies the user name used for SMTP authentication.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.39 SSOToken

Initial value:

This parameter specifies the token value to verify connection between Console and Server using SSO (single sign on) such as Windows authentication, SSO authentication. This value must be the same as the SSOToken value in the Console configuration file (`Console.conf`).

This is a dynamically reloadable property, that is, you do not need to re-start the Interstage BPM server when you change this property. For more details refer *Dynamically Reloadable Properties* on page 41.

## 4.40 SupportGroupWorkItem

Initial value: `false`

This parameter specifies support for the Group Work Item feature of Interstage BPM. If this parameter is set to `true`, you can specify work items as group work items. This means that anyone in the specified group can work on them.

**Note:**

- Note the following when setting the parameter to `true`:
  - The functionality and filtering of work items is restricted. Refer to the *API Javadoc* for details.
  - The parameter has no effect on work items associated with Voting User Task Nodes and when a Role Java Action is used. A group work item is not generated; instead, work items for each user in the group are generated.
  - In the Directory Service, don't use the same name for a user and a group. A user having the same name as a group can work on a group work item even if the user doesn't belong to that group.
- If you specify any other value than `true`, `TRUE`, `false` or `FALSE`, the Interstage BPM Server uses the initial value `false`.

## 4.41 SupportLogBufferSize

Initial value: 200

Maximum value allowed: 10000

This parameter specifies the maximum size of the support-cache used for `Support.log` in Kbytes. When the cache becomes full, it removes messages in First-In-First-Out fashion.

Note that the cache size is different from the `Support.log` file size, which is specified by the `SupportLogRecycleSize` parameter. For details of that parameter, refer *SupportLogRecycleSize* on page 56.

For more details about this support-cache and its use for `Support.log`, refer *Configuring Log Files* on page 39.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.42 SupportLogRecycleSize

Initial value: `10000`

This parameter specifies the maximum size of the `Support.log` file in Kbytes. If the log file reaches the specified size, it will be closed and renamed by adding a time stamp. A new log file with the original log file name will be opened and logging will continue with the new log file. The renamed log files can then be removed or deleted as appropriate without stopping the Interstage BPM Server. The `LogFileRecycleNumber` parameter specifies the maximum number of times a log file can be recycled.

For example if you set `SupportLogRecycleSize` as follows:

```
SupportLogRecycleSize=10000
```

Then, any log file that reaches the size of 10000 KB (10 MB) will be closed and renamed by adding a time stamp.

If the parameter is not set, the default value 10000 is used.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.43 SupportLogThreshold

Initial value: `ERROR`

If an error condition occurs, whose log level is above the threshold log-level specified by this parameter, the support-cache will flush all its messages to `Support.log`.

For details about log-levels and `Support.log`, refer *Configuring Log Files* on page 39.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.44 SuspendEventWhenProcessInError

Initial value: `true`

This parameter controls whether the state of unhandled events is set to "ignored" or "suspended" when a process instance becomes error state because of an error in processing.

This means that you can decide that the unhandled events are handled or not when the process instance gets active. The events in "ignored" state aren't handled any longer, so that you have to handle them if necessary.

- If the parameter is set to `true`, the state of unhandled events is set to "suspended" when the process instance becomes error state. In this case, the unhandled events will be handled after the node in error state gets active or the error process instance is recovered from error state to use the structured edit mode.
- If the parameter is set to `false`, the state of unhandled events is set to "ignored" when the process instance becomes error state. In this case, the unhandled events won't be handled systematically.



**Note:** This parameter is provided for the compatibility of the earlier versions of Interstage BPM Server. Set `false` to the `SuspendEventWhenProcessInError` parameter if you expect the same behavior as the earlier versions which didn't support this parameter. The default behavior has been changed.

## 4.45 TraceLogLevel

Initial value: `ALL:WARN,SYSTEM:INFO`

This parameter specifies the default log level and log category for `Trace.log`. Errors having severities equal to and above the specified log level will be logged in the file.

This parameter has the format `<LogCategory>:<LogLevel>`. You can also specify multiple levels for different categories using comma-separated values, such as

`<LogCategory>:<LogLevel>,<LogCategory>:<LogLevel>`.

The value `ALL:WARN,SYSTEM:INFO` specifies that the following errors will be logged to `Trace.log`

- Errors of all categories, with log level `WARN` or above (that is, `WARN`, `ERROR`, `FATAL`)
- Errors of `SYSTEM` category, with log level `INFO` or above (that is, `INFO`, `WARN`, `ERROR`, `FATAL`)

(Note that `ALL` is not a pure category, but can be used as a keyword to refer to all existing categories.)

**Note:** Configuration for other category will override the configuration for `ALL` category.

For more details about log levels and categories, refer *Configuring Log Files* on page 39.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.46 TraceLogRecycleSize

Initial value: `10000`

This parameter specifies the maximum size of the `Trace.log` file in Kbytes. If the log file reaches the specified size, it will be closed and renamed by adding a time stamp. A new log file with the original log file name will be opened and logging will continue with the new log file. The renamed log files can then be removed or deleted as appropriate without stopping the Interstage BPM Server. The `LogFileRecycleNumber` parameter specifies the maximum number of times a log file can be recycled.

For example if you set `TraceLogRecycleSize` as follows:

```
TraceLogRecycleSize=10000
```

Then, any log file that reaches the size of 10000 KB (10 MB) will be closed and renamed by adding a time stamp.

If the parameter is not set, the default value 10000 is used.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.47 WSCallTimeOut

Initial value: `120`

This parameter specifies the maximum time in seconds which should be permitted for response of a Web Service call. You can increase the timeout value depending on your usage requirements, for example, to 300 seconds.

If the parameter is not set, the default value 120 is used.

## 4.48 WSHttpProxyHost

Initial value:

This is the host name of your Proxy Server. You will need to configure Interstage BPM to use the Proxy Server if you are using Web Services Java Actions to call external Web Services and access the Internet only through a Proxy Server. To configure Interstage BPM to use the Proxy Server, you must set this parameter to the host name of your Proxy Server and `WSHttpProxyPort` to the port number of your Proxy Server.

Typically, a Proxy Server blocks calls to external Web Services. If you are trying to call external Web Services with a Web Service Java Action, and your Proxy Server is blocking those calls, your Web Service Java Action will fail to make a connection through the Internet, and the Java Action will fail to work properly.

Once you have performed this configuration, only Web Services that your Proxy definition allows can be called.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 4.49 WSHttpProxyPort

Initial value:

This is the port number of your Proxy Server. You will need to configure Interstage BPM to use the Proxy Server if you are using Web Services Java Actions to call external Web Services and access the Internet only through a Proxy Server. To configure Interstage BPM to use the Proxy Server, you must set this parameter to the port number of your Proxy Server and `WSHttpProxyHost` to the host name of your Proxy Server.

Typically, a Proxy Server blocks calls to external Web Services. If you are trying to call external Web Services with a Web Service Java Action, and your Proxy Server is blocking those calls, your Web Service Java Action will fail to make a connection through the Internet, and the Java Action will fail to work properly.

Once you have performed this configuration, only Web Services that your Proxy definition allows can be called.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

---

## 5 Properties of Interstage BPM Server Tenants

The configuration parameters of the Interstage BPM Server define its operating environment. They define the server's operation modes, map the locations of error logs, command scripts, and configuration files. Every tenant using Interstage BPM has properties that are unique to that tenant. These properties are called Tenant Properties. This chapter covers only the Tenant Properties. The System Properties for the Interstage BPM Server are covered in the previous chapter.

Tenant Properties allow tenants to have their own storage for external systems like a Directory Service or a Metadata Repository. It is necessary for tenants to have their own storage for these external systems to prevent other tenants from viewing this information.

Many of the tenant's configuration parameters are similar to parameters used in a query. LDAP information, for example, specifies which attributes of a user, resource, or document will be used by Interstage BPM to query the Directory Service. By providing this framework for creating queries, the configuration provides the means by which many different functions can be implemented. Access control, for example, is possible because Interstage BPM's internal mechanisms can be coordinated with the mechanism used by the external Directory Service.

The tenant's configuration parameters are stored in the Interstage BPM database. Some of the parameters were set when you carried out the initial system configuration using the Interstage BPM setup script.

If you need to reconfigure the server or change parameters that cannot be configured in the setup script, you can change the tenant configuration manually. For instructions, refer to section *Changing Parameter Values Using Tenant Management Console* on page 24 and section *Using the Configuration Export and Import Scripts* on page 25.

**Note:** Parameter names and values are case sensitive.

Be careful with any configuration changes. If you specify incorrect values, the Interstage BPM Server or other components may not be able to start or errors may occur at runtime.

This appendix lists the names and values of parameters that you may change or add. In addition, some parameters are listed that you must not change but that you may need to know about. Parameters that are used for internal purposes only are not listed. Many parameters have initial values provided by the server. These initial values, if any, are listed at the beginning of each parameter description below. You can override initial values by editing the parameter's value.

### 5.1 AdminRole

Initial value: `AdminRole`

**Note:** This parameter should not be removed and its value should not be blank.

This parameter stores the name of the group that will have tenant administrator (also known as the tenant owner) functionality, such as that of managing tenant users, process definitions, process instances and so on.

You can change the administrator to any group defined in the Directory Service or local group store. If this group is not defined in the Directory Service or local group store, or is defined with no members, no users will be able to log in for administrative functions.

### 5.2 AuditLogRecycleSize

For details, refer *AuditLogRecycleSize* on page 45.

### 5.3 DDFrameWorkImplClass

Initial value: `com.fujitsu.iflow.ddframeworkee.impl.AuthenticateHelper`

**Note:** This parameter should not be removed and its value should not be blank.

This parameter specifies the class used for the Framework Adapter.

If the parameter is not set, the default value

`com.fujitsu.iflow.ddframeworkee.impl.AuthenticateHelper` is used.

You must edit the parameter if you want to use a custom adapter.

### 5.4 DirectoryImplClass

Initial value: `com.fujitsu.iflow.diradapter.jndiimpl.DirSessionImpl`

This parameter specifies the class used for the adapter to the Directory Service. Its value depends on the adapter specified when setting up the Interstage BPM Server.

`com.fujitsu.iflow.diradapter.jndiimpl.DirSessionImpl` is used for Oracle Directory Server Enterprise Edition and for Microsoft Active Directory.

You must edit the parameter if you want to use a custom adapter.

If the parameter is not set, the default value

`com.fujitsu.iflow.diradapter.jndiimpl.DirSessionImpl` is used.

### 5.5 DirectoryService

Initial value: `0`

This parameter specifies what Directory Service Interstage BPM uses:

- `0`=Local User Management
- `1`=JNDI (LDAP/Oracle Directory Server Enterprise Edition)
- `2`=Custom Adapters
- `3`=AD (Microsoft Active Directory)

**Note:** If the parameter is not set or if it is set to an invalid value, the default value `0` is used.

### 5.6 EmailListenerAutoReplyEnabled

Initial value: `true`

This parameter enables or disables the Interstage BPM Server sending out a message in response to the receipt of the task completion email message sent by the task assignee or in response to the receipt of the email message containing Trigger Payload sent by a user. It can be set to `true` or `false`.

### 5.7 EmailListenerDeleteInvalidMessages

Initial value: `true`

This parameter enables or disables the deletion of invalid task completion email messages. Invalid messages are those that the Email Listener/Server cannot recognize, so they cannot be processed. An example of an invalid message would be a task completion message that attempts to make a

choice on an already completed task. Deleting the invalid email messages reduces the load on the Server. It can be set to `true` or `false`.

## 5.8 EmailListenerEmailAddress

Initial value:

The email address of the Email Listener. This is the address to which task completion email messages and the email messages with Trigger payload are sent. It will automatically appear in the **To** field of the message. This property must be set to a valid email address that is associated with a valid email account on an email Server that can be accessed using the POP email protocol.

## 5.9 EmailListenerEnabled

Initial value: `false`

This parameter enables or disables the Email Listener for a tenant. If this property is set to `false`, users can still receive the Email Notifications even though the listener is disabled. It can be set to `true` or `false`.

## 5.10 EmailListenerPassword

Initial value:

This is the password needed to access the mail store.

## 5.11 EmailListenerPollingInterval

Initial value: 900

**Note:** This value is in seconds. 900 seconds is equivalent to 15 minutes.

This is the polling interval for the Email Listener. The polling interval is the interval of time upon which the Email Listener checks for new messages.

## 5.12 EmailListenerPOPPort

Initial value: 110

This is the port number used by the POP protocol for the Email Listener.

## 5.13 EmailListenerPropertiesFile

Initial value:

This properties file is used to set multiple properties if they are needed to access the mail server using POP3. For instance if the mail server is SSL enabled, then a few optional properties may be set. The Super User can use this properties file to set any additional properties that are required. The list of optional properties is available in the *JavaMail API specification* provided by Oracle Corporation.

## 5.14 EmailListenerServerHost

Initial value:

---

This is the Host Name of the mail server used to send messages from the Interstage BPM Server.

## 5.15 EmailListenerUserName

Initial value:

This is the user name needed to access the mail store.

## 5.16 EmailNotificationEnabled

Initial value: `true`

This parameter enables or disables the email notification feature. It can be set to `true` or `false`.

**Note:** If the parameter is not set or if it is set to an invalid value, the default value `true` is used.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.17 EmailStyleSheetFile

Initial value:

This property specifies the HTML style sheet (includes path to the file) used for the task assignment email message and response email messages that the Interstage BPM Server sends out. The Interstage BPM Server applies the styles specified in the style sheet to this message.

## 5.18 ErrorLogLevel

For details, refer *ErrorLogLevel* on page 47.

## 5.19 ErrorLogRecycleSize

For details, refer *ErrorLogRecycleSize* on page 48.

## 5.20 IASSOLoginConfFileEntryName

Initial value:

This parameter specifies what SSO (single sign on) Interstage BPM uses.

A blank value indicates no SSO will be used.

The value of `com.fujitsu.iflow.sso` indicates Windows Authentication / built-in SSO authentication will be used.

## 5.21 IBPMUserStoreActive

Initial value: `true`

This parameter specifies whether Interstage BPM's local user store is used to manage Interstage BPM user accounts and authenticate users logging in to Interstage BPM:

- If the parameter is set to `true`, the local user store is used.
- If the parameter is set to `false`, the Directory Service specified in the `DirectoryService` parameter is used.

This parameter only defines the system behaviour regarding user management and user authentication. Groups can always be managed in Interstage BPM's local group store regardless of how this parameter is set.

**Note:**

- If the parameter is not set or if it is set to an invalid value, the default value `true` is used.
- The value of this parameter controls the initial value of the `RequiredUserAttribute` parameter. For information, refer *RequiredUserAttribute* on page 67.

## 5.22 LDAPAccessUserID

Initial value:

This is the user ID required to create the base structure in the Directory Service for Interstage BPM. It is also required by Interstage BPM during operation to access the Directory Service for authentication and Interstage BPM user profile handling. The value is set when you setup the Interstage BPM Server.

## 5.23 LDAPAccessUserPassword

Initial value:

This is the password used in conjunction with `LDAPAccessUserID`. It is required by the Interstage BPM setup script to create the base structure in the Directory Service for Interstage BPM, and required by Interstage BPM during operation to access the Directory Service for authentication and Interstage BPM user profile handling.

The password is set when you setup the Interstage BPM Server and it is encrypted for security reasons. If you want to change the password, you can use the `TextEncrypter` utility to encrypt the value. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

## 5.24 LDAPContextFactory

Initial value: `com.sun.jndi.ldap.LdapCtxFactory`

This parameter specifies the name of the LDAP Context Factory class. The class is used to authenticate users against a Directory Service. Do not edit this parameter.

## 5.25 LDAPDBSuffix

Initial value:

This parameter specifies the directory sub-tree in the Directory Service where groups and users to be used with Interstage BPM are stored. Check with your LDAP administrator for the values for the directory sub-tree for your LDAP setup. This value is set when you setup the Interstage BPM Server.

**Note:** Users are stored under the "People" organization unit, and groups are stored under the "Groups" organization unit. As such, Interstage BPM prepends the value of this parameter with `ou=people` when searching for users and `ou=groups` when searching for groups.

### Example:

If you specify `LDAPDBSuffix=o=fujitsu.com,c=us`, then Interstage BPM will search for users under `ou=people,o=fujitsu.com,c=us` and for groups under `ou=groups,o=fujitsu.com,c=us`

## 5.26 LDAPFilter

Initial value:

The setting of this parameter to `cn=*` as a search filter indicates that Interstage BPM wants to retrieve all user IDs for authentication of logins and other authentication purposes. This is consistent with the requirements of most LDAP servers, but others may require a different filter keyword. The `LDAPFilter` parameter can be used to specify an alternate search filter notation to be used by Interstage BPM in communicating with the authentication service for Interstage BPM user IDs. This value is set when you setup the Interstage BPM Server, and should not be changed.

## 5.27 LDAPGroupCacheAgeSec

Initial value: 3600

This parameter is specified as some number  $x$ , where  $x$  is in seconds. The initial value is 3600, which means that the LDAP Group Cache expires and new LDAP Group information is cached every hour.

**Note:** If this parameter is not set or if you specify a value other than a number, the Interstage BPM Server uses the initial value 3600.

## 5.28 LDAPGroups

Initial value:

This parameter supports an LDAP implementation used in previous versions of the Interstage BPM software and is only used if `LDAPUseDefUsersGroups=Yes`. The `Groups` ou with default Interstage BPM Groups should be added by using the Directory Server's tools. The `LDAPGroups` parameter can be used to specify an alternate organizational unit and name to be used by Interstage BPM in communicating with the authentication service for Interstage BPM role names.

## 5.29 LDAPGroupsDN

Initial value:

This parameter specifies the fully-qualified domain name of the LDAP Group that defines a particular set of Interstage BPM roles. This is the set of Interstage BPM roles that are taken directly from the set of Groups defined in LDAP before the setup of Interstage BPM. The `IflowGroups` LDAP Group should be added by using the Directory Server's tools. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for more information and instructions.

## 5.30 LDAPPeople

Initial value:

This parameter supports an LDAP implementation used in previous versions of the Interstage BPM software and is only used if `LDAPUseDefUsersGroups=Yes`. The `People` ou with default Interstage BPM users should be added by using the Directory Server's tools. The `LDAPPeople` parameter can be used to specify an alternate organizational unit and name to be used by Interstage BPM in communicating with the authentication service during session login.

## 5.31 LDAPServer

Initial value:

---



This parameter identifies the server (machine name) that the LDAP server is installed on. This value is set when you setup the Interstage BPM Server.

**Note:** Do not change the port number, even if you move the server.

## 5.32 LDAPuid

Initial value:

This parameter is only relevant if you are using Active Directory or Oracle Directory Server Enterprise Edition. It specifies the name of the attribute that identifies the user ID within an LDAP login string. Depending on the parameter value, Interstage BPM passes the following string to the Directory Service:

- `cn=<user ID>` with Active Directory
- `uid=<user ID>` with Oracle Directory Server Enterprise Edition

Add the `LDAPuid` parameter to the configuration parameters and specify the name that needs to be passed to your Directory Service.

Example: `LDAPuid=CN`

## 5.33 LDAPUseDefUsersGroups

Initial value:

This parameter supports an LDAP implementation used in previous versions of the Interstage BPM software to support backward compatibility. When this parameter is set to `Yes`, it will use the `LDAPGroups` and `LDAPPeople` parameters as specified in their descriptions and assume that you have the organizational units (`ou`'s) defined in them. For example if you use Interstage BPM with LDAP as setup, this parameter will be set to `Yes`. `LDAPGroups` will be set to `ou=Groups` and `LDAPPeople` will be set to `ou=People`. Interstage BPM will assume that you have defined Groups and People organizational units and use them.

If this parameter is set to `No`, Interstage BPM will use the following groups and users:

- Users:
  - Member users of a group specified in the `LDAPUsersDN` parameter.
  - Member users of groups (including sub-level groups) specified in the `LDAPGroupsDN` parameter.
- Groups:
  - Member groups of groups specified in the `LDAPGroupsDN` parameter.

## 5.34 LDAPUsersDN

Initial value:

This parameter specifies the fully-qualified domain name of the LDAP Group that defines a particular set of Interstage BPM users. This is the set of users that are taken directly from the set of users defined in LDAP before the setup of Interstage BPM. The `IflowUsers` LDAP Group should be added by using the Directory Server's tools. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for more information and instructions.

## 5.35 LogFileRecycleNumber

For details, refer *LogFileRecycleNumber* on page 49.

## 5.36 ManagerAttribute

Initial value: `manager`

The value of this parameter specifies the name of the attribute that will hold the user ID of a user's manager. While the initial value of this parameter is `manager`, it can be changed to any other name that indicates a supervisory reporting relationship. For example, you can change this to `ReportingManager` or `Supervisor` as per your organization's requirements.

The value specified here should also be included in the `RequiredUserAttribute` parameter, with the `cache` option. For example, if the `ManagerAttribute` parameter has the value `ReportingManager`, include `ReportingManager:cache` in the `RequiredUserAttribute` parameter.

Also refer *RequiredUserAttribute* on page 67 and *TeamAttribute* on page 72.

To be able to retrieve organization hierarchy information, and use it to get information, (for example, about a user's subordinates and their work item status) you must specify either of `ManagerAttribute` or `TeamAttribute` properties.

Use only either of `ManagerAttribute` or `TeamAttribute` properties, not both. If both properties are specified, only the `TeamAttribute` property is used.

## 5.37 MetadataRepositoryPassword

Initial value:

This is the password of the user used for publishing process definition information to a WebDAV Document Management System (DMS) associated with Interstage BPM. The user associated with this password must have the administrative rights to publish to the WebDAV Repository.

Do not specify a Metadata Repository Password if no WebDAV-compliant system is associated with Interstage BPM.

The password is encrypted for security reasons. If you want to change the password, you can use the `TextEncrypter` utility to encrypt the value. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

## 5.38 MetadataRepositoryURL

Initial value:

This is the location used for publishing process definition metadata to a WebDAV Document Management System (DMS) associated with Interstage BPM.

Do not specify a Publish Metadata URL if no WebDAV-compliant system is associated with Interstage BPM.

## 5.39 MetadataRepositoryUserName

Initial value:

This is the user name used for publishing process definition information to a WebDAV Document Management System (DMS) associated with Interstage BPM. The user specified must have the administrative rights to publish to the WebDAV Repository.

Do not specify a Metadata Repository User Name if no WebDAV-compliant system is associated with Interstage BPM.

## 5.40 RequiredUserAttribute

Initial value: depends on the value of the `IBPMUserStoreActive` property

- If `IBPMUserStoreActive` is `true`, initial value of `RequiredUserAttribute` is `uid:local,iflownotification:local,email:local,dmsdir:local,manager:local:cache,favorites:local,dashboard:local,cn:local:cache,title:local:cache`
- If `IBPMUserStoreActive` is `false`, initial value of `RequiredUserAttribute` is `uid:local,iflownotification:local,email:local,dmsdir:local,manager:cache:dn,favorites:local,dashboard:local,cn:cache,title:cache`

**Note:** Values above should be used without spaces.

This parameter specifies the default user attributes and their default options that Interstage BPM uses for the User Profile feature. You can add new attributes to this list, separated by commas. Make sure that the newly added attributes are defined in the schema of your Directory Service or in Interstage BPM's local user store.

Only attributes specified in the `RequiredUserAttribute` parameter are used for the Interstage BPM User Profile feature. All of the attributes specified here can be edited through the Interstage BPM Console.

Each attribute can have the following options:

- `local`: specifying this causes the values of the attribute to be stored in the Interstage BPM database, even if you use a remote directory server such as LDAP.
- `cache`: specifying this causes the values of the attribute to be stored in the cache of the Interstage BPM database, thus enabling you to use search functionality on this attribute.

**Note:** Using this option brings in the following length restrictions on the attribute:

- Its name should not exceed 64 characters.
- Its value should not exceed 256 characters.

If these limits are exceeded, the name and values will not be stored in the cache.

Interstage BPM Server refreshes the user attributes in the cache in the following cases:

- When you modify the `RequiredUserAttribute` parameter values
- When you call the Model API `WFAdminSession.refreshAllUserProfileCache()` method.
- `dn`: You can specify this option only for attributes whose values are user names (for example, the `uid` and `manager` attributes, whose values are user names). If you specify the `dn` option for an attribute:
  - Interstage BPM Server converts the attribute's values (user names) to the `DN` (Distinguished Name) format before storing it in the directory server
  - While retrieving the attribute values, Interstage BPM Server converts the values from the `DN` format to user names.

**Note:** If your user attribute values are stored in a remote directory server, note the following:

- Oracle Directory Server Enterprise Edition can store multiple values of the `manager` user attribute
- Microsoft® Active Directory can store only a single value of the `manager` user attribute

Also refer *IBPMUserStoreActive* on page 62, *ManagerAttribute* on page 66 and *TeamAttribute* on page 72.

## 5.41 SecuritySwitch

Initial value: `false`

Interstage BPM can be set to two security levels: Demo Mode and Security Mode. Demo Mode is useful while key members of your work team learn the features of the software. Later, Security Mode provides a stable and controlled environment for implementing Interstage BPM across your company.

**Note:** If you specify any other value than `true`, `TRUE`, `false` or `FALSE`, the Interstage BPM Server uses the initial value `false`.

### Demo Mode (`false`)

This mode restricts access to Interstage BPM in two ways:

- Only the person who is assigned a work item may make choices about it, thus advancing the process to the next work item - or completing it.
- Only members of the `AdminRole` can log in to a client as an Admin user to access the Administration functionality in the Model API.

In all other ways, all Interstage BPM users can experiment with all of its functions, including creating and editing process definitions and instances while it is running in Demo Mode.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

### Security Mode (`true`)

In Security Mode, access to process definitions, process instances, and work items is restricted to the specific people who need to use them.

- Only members of the group specified in the `TemplatePublishers` parameter can create new process definitions.
- Only the owner of the process definition can edit it.
- Only the process owners and current assignees can view any specific process. (“Current assignees” are those who have active task assignments. This does not include people who will have task assignments later in the process.)
- Only the process owners can structurally edit their specific process. This means they can add, modify or delete structural elements of the process instance like node instances, arrow instances and User Defined Attributes.
- Only assignees of an activity can commit (i.e., complete and sign off) the work item.

## Setting Up Security Mode

To turn on Security Mode:

1. Specify `SecuritySwitch=true` to enable Security Mode.
2. Also specify `TemplatePublishers=<group name>`. This group must be defined in your Directory Service or in Interstage BPM's local group store. For example, if you have defined a group of `pubRole` in the Directory Service or local group store, you would type `TemplatePublishers=pubRole`.

## 5.42 SendAnalyticEvents

Initial value: `false`

This parameter specifies whether the Interstage BPM Server sends events to a JMS queue that can then be evaluated by Interstage Analytics. The parameter can be set to `true` or `false`.

**Note:** If the parameter is not set or if it is set to an invalid value, the default value `false` is used.

## 5.43 ServerEmailAddress

Initial value: `<email address specified during tenant creation>`

This parameter specifies the address that will appear in the `From:` field of all email messages sent from the Interstage BPM Server. This includes all email notifications, work item escalations, timer resolution messages, Email Listener, etc.

**Note:** To successfully use these abilities, this parameter should not be blank.

This email address should be the email address or alias email address of the Interstage BPM Super User that manages the email server used by Interstage BPM.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.44 ServerEmailClientPage

Initial value:

The value of this parameter is appended to the value of the `ServerEmailBaseURL` parameter to generate a URL that points to the email client. If you develop a custom email client, specify the appropriate value for this parameter.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.45 ServerLogsDirectory

For details, refer *ServerLogsDirectory* on page 53.

## 5.46 ServerNotificationUserProfileField

Initial value: `iflownotification`

The parameter specifies the name of the field associated with the user in the Directory Service or in Interstage BPM's local user store, which in turn specifies whether the user is to receive email notification or not.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.47 ServerPassword

Initial value: <Server Login password for tenant owner>

**Note:** This parameter should not be removed and its value should not be blank.

This is the password the Interstage BPM Server uses for logging into the Directory Service to ascertain directory settings. The login takes place when the Interstage BPM Server starts up.

`ServerUserName` specifies the tenant owner associated with this password.

The password is set when you create the tenant owner while setting up the Interstage BPM Server and it is encrypted for security reasons. If you want to change the password, you can use the `TextEncrypter` utility to encrypt the value. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

**Note:** If Local User Management (LUM) is used as directory service then the change in tenant owner password will automatically change the `ServerPassword`. Refer section *Changing Username and Password of Tenant Owner of Interstage Business Process Manager Server and Console Installation Guide* for details.

## 5.48 ServerReassignMode

Initial value: `regular`

This field is used to set Interstage BPM's Reassignment Mode to one of the following:

- **Regular Mode** (`regular`): Any current assignee of the activity or any of its process owners can reassign the activity to new assignees.
- **Process-Owner-Only Mode** (`owner`): Only a process owner can reassign an activity to one or more other people.
- **No-Reassignment Mode** (`none`): Reassignment is disabled, and nobody can reassign an activity.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.49 ServerUserName

Initial value: <Server Login user id of tenant owner specified during tenant creation>

**Note:** This parameter should not be removed and its value should not be blank.

This is the user name the Interstage BPM Server uses for logging into the Directory Service to ascertain directory settings. The login takes place when the Interstage BPM Server starts up.

You specified the user name of the tenant owner when setting up the Interstage BPM Server. The user specified must have access to system resources.

## 5.50 SSOAuthentication

Initial value:

This parameter enables or disables the built-in SSO(Single Sign On) authentication for a tenant. This authentication process allows the user to log in once and gain access to multiple (SSO enabled) applications without having to login again in each application during that session.

**Pre-requisites:**

- Interstage BPM OpenID Provider application is started.
- The tenant uses Microsoft Active Directory servers or Oracle Directory Server Enterprise Edition. (The `DirectoryService` property of Interstage BPM Server Tenant must be set to 1 or 3).
- The OpenID Provider application and tenant both have the same LDAP configuration.

**Note:** This parameter is used by Interstage BPM Console only and can not be changed using the configuration export and import scripts as described in the section *Using the Configuration Export and Import Scripts* on page 25.

This parameter can be updated as mentioned below:

- **Using the Interstage BPM Tenant Management Console**

Follow these steps to enable/disable the SSO authentication:

1. Login to the Tenant Management Console at the following URL:

```
http://<FQDN>:<port>/console/TenantManager.page
```

2. Select the tenant for which you want to enable/disable the SSO Authentication and click the **Properties** tab.
3. Navigate to the **SSOAuthentication** property and do the following:
  - **To enable the SSO authentication:** Check the **Use Built-in SSO** check box.

The value for **OpenID Provider URL** gets auto-populated. Its value must be in the following format:

```
[http|https]://<FQDN>:<port>/<Provider context-root name>/id/
```

Here,

- `<FQDN>` is the fully qualified domain name (machine name / IP address) on which OpenID Provider application is setup.

**Note:**

- Make sure you specify the `<FQDN>` that can be resolved by the machines accessing Console application in a web browser; otherwise error page might be displayed.
- If you modify the `<FQDN>` value (for example - from IP address to machine name) for same OpenID Provider, then you must restart the OpenID Provider application.
- In secure mode, ensure that the machine name / IP address you are specifying here exists as the value of `Common Name` in the security certificate. Otherwise, create a new certificate.

- `<port>` is the port number configured for OpenID Provider application.
- `<Provider context-root name>` is the context-root name of the OpenID Provider application.

**Note:** If you use the OpenID provider application that is setup by the Interstage BPM setup script, use the following values:

- `<FQDN>`: This value should be same as fully qualified domain name of Interstage BPM Console.
- `<port>`: This value should be same as port number of Interstage BPM.
- `<Provider context-root name>`: This value should be `openid`.

Example: `http://<FQDN>:<port>/openid/id/`

- **To disable the SSO authentication:** Uncheck the **Use Built-in SSO** check box.

4. Click **Save**.

## 5.51 SWAPLinkagePassword

Initial value: `<Server Login user password>`

This parameter specifies the password of the user that actually initiates subprocesses requested by process instances running on remote servers.

You can use the `TextEncrypter` utility to encrypt the value. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.52 SWAPLinkageUserName

Initial value: `<Server Login user id>`

This parameter specifies the user that actually initiates subprocesses requested by processes running on remote servers.

Refer to section *Integrating Interstage BPM with Other Servers* on page 32 for more information about the Interstage BPM Linkage User.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.53 TeamAttribute

Initial value:

The value of this parameter specifies the name of the attribute that will hold the user IDs of a user's team members who report to that user. While this parameter has no initial value, you can add any name that indicates a subordinate reporting relationship. For example, you can add `TeamMember` as a value for this parameter.

The value specified here should also be included in the `RequiredUserAttribute` parameter, with the `cache` option. For example, if the `TeamAttribute` parameter has the value `TeamMember`, include `TeamMember:cache` in the `RequiredUserAttribute` parameter.

Also refer *RequiredUserAttribute* on page 67 and *ManagerAttribute* on page 66.

To be able to retrieve organization hierarchy information, and use it to get information (for example, about a user's subordinates and their work item status) you must specify either of `ManagerAttribute` or `TeamAttribute` properties.



Use only either of `ManagerAttribute` or `TeamAttribute` properties, not both. If both properties are specified, only the `TeamAttribute` property is used.

## 5.54 TemplatePublishers

Initial value: `IFlowPublishers`

This parameter is specified in conjunction with enabling Security Mode by setting the `SecuritySwitch` parameter. The value of this parameter is the name of the group defined in the Directory Service or in Interstage BPM's local group store, which contains a list of members who are authorized to publish process definitions while in Security Mode.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 5.55 TraceLogLevel

For details, refer *TraceLogLevel* on page 57.

## 5.56 TraceLogRecycleSize

For details, refer *TraceLogRecycleSize* on page 57.

## 5.57 UDDIBusinessName

Initial value:

This is the name of the organization that is publishing business process definition information to the UDDI Registry. It is an identifier that helps you to categorize the process definition information added to the UDDI Registry and makes it easier to search and retrieve the process definition Web Services.

In order to be able to publish process definition information to the UDDI Registry, the name specified in `UDDIBusinessName` must exist in the UDDI Registry.

Do not specify a UDDI Business Name if no UDDI Registry is associated with Interstage BPM.

## 5.58 UDDIInquiryURL

Initial value:

This parameter specifies UDDI Query Server URLs. UDDI Query Server URLs are used to search UDDI Registries containing information about Web Services (WSDL documents). Multiple URLs may be added to this parameter in a comma-separated list.

## 5.59 UDDIPublisherPassword

Initial value:

This is the password of the user used for publishing process definition information to a UDDI Registry associated with Interstage BPM. `UDDIPublisherUserName` specifies the user associated with this password.

Do not specify a UDDI Publisher Password if no UDDI Registry is associated with Interstage BPM.

The password is encrypted for security reasons. If you want to change the password, you can use the `TextEncrypter` utility to encrypt the value. Refer to the *Interstage Business Process Manager Server and Console Installation Guide* for details.

## 5.60 UDDIPublisherUserName

Initial value:

This is the user name used for publishing process definition information to a UDDI Registry associated with Interstage BPM. The user specified must have the following rights:

- Administrative rights to publish to the UDDI Registry
- Rights to publish data to the organization specified in `UDDIBusinessName`

Do not specify a UDDI Publisher User Name if no UDDI Registry is associated with Interstage BPM.

## 5.61 UDDIPublishURL

Initial value:

This is the location used for publishing process definition information to a UDDI Registry associated with Interstage BPM.

Do not specify a UDDI Publish URL if no UDDI Registry is associated with Interstage BPM.

## 5.62 WebDAVKeyStore

Initial value:

This parameter should point to the path of the keystore, where the SSL security certificate obtained from WebDAV Server is added.

## 5.63 WIResponsible

Initial value: `User`

This parameter specifies the permission level for acting on a work item. The available options for this parameter are:

- `User`: This is default setting. Only a work item assignee of the work item can make a choice on the work item.
- `Group`: Any member of a group can make a choice on work items assigned to any other user in the group.
- `All`: Anyone can make a choice on any work item.

This parameter is dynamically reloadable. For more information, refer *Dynamically Reloadable Properties* on page 41.

## 6 Making a Backup of Your Data

As a system administrator, back up the following data against an unexpected system shut down:

- Configuration Files
- Log Files
- Document Management System (DMS) directory
- Database

Refer to the following sections for more backup-related information.

### 6.1 Configuration Files

Besides the configuration parameters stored in the Interstage BPM database, Interstage BPM uses several configuration files. The following table gives a list of the configuration files, their typical location and their meaning.

It is assumed that Interstage BPM is setup in C:\Fujitsu\InterstageBPM on Windows and in /opt/FJSVibpm on UNIX.

Configuration File	Location	Remarks
DmsCollections.xml	<engine directory>/server/ instance/default/tenants/ <tenant name>/resources	Configuration file for the Document Management System (DMS) for a tenant.
kpilistener.conf	<engine directory>/server/ instance/default/resources	Configuration file for the KPI Listener.

### 6.2 Log Files

For details of log files, refer *Configuring Log Files* on page 39.

### 6.3 DMS Directory

Make sure that you backup all directories and sub-directories defined in `DmsCollections.xml`; they include all files uploaded by users, such as Workflow Applications, located in the `<DMSRoot>/apps` directory.

### 6.4 Database

All data, which result with the work with the Interstage BPM, for example process definitions and process instances, are stored in the database. You have to do database operations like the following with the default functionalities of the database server that you are using:

- Integrity check
- Database statistics
- Database backups
- Restoring databases

Refer to the manuals of your database server for more information.

## Appendix A: Database Schema

**Note:** This Appendix serves only for information purposes. The Database Schema described in this Appendix could be changed in a future version of Interstage BPM. If you develop your own application which is based on the Database Schema, you need to change the application when the Database Schema is changed in future.

The following sections describe all tables currently available.

### ActiveUser

This table represents the data about a logged in user.

Column Name	Types	Remarks
id	NUMBER NOT NULL	Each logged in user is given a unique identifier by the server.
longRaw	BLOB NOT NULL	The data associated with the user.
serverName	NVARCHAR2(256) NOT NULL	The name of the server where the user logged in.
tenantId	NUMBER NOT NULL	Unique identifier of a tenant
PRIMARY KEY(ID)		Each user is distinguished by the identifier assigned to it by the server.

FOREIGN KEY (tenantId) references Tenant(tenantId)

### ActivityDataSet

This table represents the data set associated with a node instance.

Column Name	Types	Remarks
activityInstanceId	NUMBER NOT NULL	The identifier of the node instance for the associated data set.
longRaw	BLOB NULL	An array of "name-type-value" structure representing the data associated with the node instance.
PRIMARY KEY(ACTIVITYINSTANCEID)		There is only one data set associated with a unique node instance.

### ActivityDataSetDefinition

This table represents the data set definitions associated with each node instance.

Column Name	Types	Remarks
name	NVARCHAR2(64) NOT NULL	Name of the data element.

Column Name	Types	Remarks
activityDefinitionId	NUMBER NOT NULL	The identifier of the node definition with which this data element is associated.
type	NUMBER NOT NULL	Type of the data element.
PRIMARY KEY(NAME, ACTIVITYDEFINITIONID)		Each data element is distinguished by its name and the identifier of the node definition with which it is associated.

FOREIGN KEY (activityDefinitionId) references ActivityDefinition(activityDefinitionId)

### ActivityDefinition

This table represents a node definition of Interstage BPM. Node definitions serve as templates for the creation of node instances of process instances.

Column Name	Types	Remarks
activityDefinitionId	NUMBER NOT NULL	The identifier of the node definition.
subProcessDefinitionId	NUMBER NOT NULL	For chained and subprocess nodes, the identifier of the subprocess spawned by this node. (DEPRECATED)
activityTypeId	NUMBER NOT NULL	The identifier depicting the type of the node definition.
label	NVARCHAR2(64) NOT NULL	The name of the node definition.
processDefinitionId	NUMBER NOT NULL	The identifier of the process definition to which this node definition belongs.
activityRole	NVARCHAR2(256) NULL	The name of the group(s) that are the valid recipients of this node instance when it is instantiated as part of the process instance.
longRaw	BLOB NULL	The data associated with this node.
posX	NUMBER NOT NULL	The x-coordinate of the lower left corner of the node definition for use with the Interstage BPM Console.
posY	NUMBER NOT NULL	The y-coordinate of the lower left corner of the node definition for use with the Interstage BPM Console.
description	NVARCHAR2(256) NULL	A detailed explanation regarding this node.

Column Name	Types	Remarks
subProcessDefinitionName	NVARCHAR2(64) NULL	For chained and subprocess nodes, the name of the process definition, from which a process instance that will be spawned by this node.
flags	NUMBER NULL	An optimization parameter recording the presence of various scripts for this node.
customNodeType	NVARCHAR2(256) NULL	String value containing the fully qualified class name of the Custom Node. If this node is not a Custom Node, then NULL is stored or returned.  With Custom Nodes, any Node may now have a custom UI presentation. While it is still based upon one of the standard node types. Custom Nodes are uniquely identified by the fully qualified class name that implements their UI.
subProcessDefinitionURI	NVARCHAR2(2000) NULL	The URI used for identifying the subprocess to be started.
priority	NUMBER NOT NULL	The priority of the node definition. This will be the default priority of node instance.
PRIMARY KEY(ACTIVITYDEFINITIONID)		Each node definition is distinguished by its unique identifier.

FOREIGN KEY (processDefinitionId) references ProcessDefinition(processDefinitionId)

### ActivityInstance

An activity instance represents a single activity in a process instance. In addition, it's the same meaning as a node instance, and described as "a node instance" in manuals of Interstage BPM.

Column Name	Types	Properties	Remarks
activityInstanceid	NUMBER NOT NULL		The identifier of the node instance.

Column Name	Types	Properties	Remarks
state	NUMBER NOT NULL	0=Initial 1=Ready 2=Disabled 3=AisRunning 4=WaitingOnSubProcess 5=AisClosed 6=AisAborted 7=AisCompleted 8=LastWorkItemCompleted 9=LastWorkItemAborted 10=AisError 11=AisSuspend	The state of the node instance.
processInstanceId	NUMBER NOT NULL		The identifier of the process instance to which this node instance belongs.
activityDefinitionId	NUMBER NOT NULL		The identifier of the node definition which is the process definition for this node.
subProcessInstanceId	NUMBER NOT NULL		For chained and subprocess node instances, the identifier of the subprocess spawned by this node.
longRaw	BLOB NULL		The data associated with this node instance.
subProcessInstanceURI	NVARCHAR2(2000) NULL		The URI used for identifying the subprocess instance to be started.
dueDateExpireTime	DATE NOT NULL		The due date time in milliseconds. If this node instance has no due date, the expire time is 01-01-1970.
priority	NUMBER NOT NULL		The priority of node instance.
label	NVARCHAR2(64) NOT NULL		The name of the node instance.

Column Name	Types	Properties	Remarks
activityTypeid	NUMBER NOT NULL	0 = Start Node 1 = End Node 2 = User Task / Service Task Node 4 = Parallel Join Gateway Node 5 = Simple / Flexible Exclusive Gateway Node 6 = Parallel Split Gateway / Send Task / Business Rule Task / Script Task / Email / DB / Web Service Node 7 = Call Activity Node 8 = Timer Node 9 = Chained-Process Node 10 = Voting User Task Node 11 = Remote Sub-Process Node 12 = Receive Task / Message Receive Node 13 = Embedded Sub-Process Node 14 = Dynamic User Task Node	The identifier depicting the type of the node instance.
PRIMARY KEY (ACTIVITYINSTANCEID)			Each node instance is distinguished by its unique identifier.

FOREIGN KEY (processInstanceid) references ProcessInstance(processInstanceid)

### ActivityTimerDefinition

This table represents the timer definitions associated with a node definition.

Column Name	Types	Remarks
timerId	NUMBER NOT NULL	The identifier of the timer definition.
activityDefinitionId	NUMBER NOT NULL	The node definition with which this timer is associated.
label	NVARCHAR2(64) NULL	The name of the timer.
data	BLOB NULL	The data associated with this timer.



Column Name	Types	Remarks
isDueDate	NUMBER NULL	Indicates that this timer definition defines a due date time calculation and related optional actions.
PRIMARY KEY(TIMERID)		Each timer definition is distinguished by its unique identifier.

FOREIGN KEY (activityDefinitionId) references ActivityDefinition(activityDefinitionId)

### AdvanceSearch

This table represents an advanced search for work items or process instances defined in the Interstage BPM Console.

Column Name	Types	Properties	Remarks
userId	NVARCHAR2(64) NOT NULL		The user who saved this advanced search. It belongs to this user and is not visible to any other user.
queryName	NVARCHAR2(64) NOT NULL	WI = Work Item PI = Process Instance	The name of the advanced search.
queryType	NVARCHAR2(64) NOT NULL	0 = not default 1 = default	The type of this advanced search.
defaultValue	NUMBER NULL		Identifies whether this is a default search for this user or not. If it is a default search, it will be run automatically when the user visits the Work Item list page or Process Instances list page in the Console.
queryString	NCLOB NOT NULL		The set of parameters associated with this search.
tenantID	NUMBER NOT NULL		Unique identifier of tenant
PRIMARY KEY(USERID, QUERYNAME, QUERYTYPE, TENANTID)			Each advanced search is distinguished by user ID, query name, query type and tenant ID.

FOREIGN KEY (tenantId) references Tenant(tenantId)

### ApplicationErrorNode

Column Name	Types	Remarks
entryNr	NUMBER NOT NULL	
applicationContainerId	NUMBER NOT NULL	Unique identifier of application that contains the error node
errorNode	NVARCHAR2(256) NOT NULL	
targetState	NUMBER NOT NULL	
PRIMARY KEY (ENTRYNR)		Each application error node is distinguished by its unique entry number.

FOREIGN KEY (applicationContainerId) references WorkflowApplication(applicationContainerId)

### ApplicationTimer

This table represents timers contained in the Process Scheduler associated with an application.

Column Name	Types	Remarks
timerInstanceid	NUMBER NOT NULL	Unique identifier of the timer instance of the process scheduler
applicationContainerId	NUMBER NOT NULL	Unique identifier of the application that contains the process scheduler
label	NVARCHAR2(64) NOT NULL	The name of the timer
data	BLOB NULL	The data associated with the timer
realTimeInMillis	NVARCHAR2(64) NOT NULL	This indicates the absolute time when this timer will expire.
lastTimeInMillis	NVARCHAR2(64) NOT NULL	This indicates the last expiration time of this timer.
isHandled	NUMBER NOT NULL	Indicates the state of the timer. 0 = NotHandled 1 =Handled 3 = Error 8 = ApplicationStopped
serverName	NVARCHAR2(64) NULL	The name of the server on which the Process Scheduler is started and handled.
PRIMARY KEY(TIMERINSTANCEID)		Each timer instance is distinguished by its unique identifier

FOREIGN KEY (applicationContainerId) references WorkflowApplication(applicationContainerId)

## ArchivedProcessDefinition

This table represents archived process definitions.

Column Name	Types	Remarks
id	NUMBER NOT NULL	The identifier of the process definition.
label	NVARCHAR2(64) NOT NULL	The name of the process definition.
xmlString	NCLOB NULL	The XML string representing the archived process definition with its associated nodes, arrows, timers and data elements.
applicationContainerId	NUMBER NOT NULL	Unique identifier for the application that contains the archived process definition
PRIMARY KEY(ID)		Each archived process definition is distinguished by its unique identifier.

FOREIGN KEY (applicationContainerId) references WorkflowApplication(applicationContainerId)

## ArchivedProcessInstance

This table represents archived process instances.

Column Name	Types	Remarks
id	NUMBER NOT NULL	The identifier of the process instance.
label	NVARCHAR2(64) NOT NULL	The name of the process instance.
processDefinitionId	NUMBER NOT NULL	The process definition from which this process instance was created.
timeCreated	DATE NULL	The time the process instance was created.
timeCompleted	DATE NULL	The time the process instance was closed.
xmlString	NCLOB NULL	The XML string representing the archived process instance with its associated node instances, arrow instances, timer instances and data elements.
applicationContainerId	NUMBER NOT NULL	Unique identifier of the application that contains this process instance
PRIMARY KEY(ID)		Each archived process instance is distinguished by its unique identifier.

FOREIGN KEY (applicationContainerId) references WorkflowApplication(applicationContainerId)

## ArrowDefinition

This table represents an arrow definition of Interstage BPM. Arrow definitions serve as templates for the creation of arrow instances of process instances.

Column Name	Types	Remarks
arrowDefinitionId	NUMBER NOT NULL	The identifier of the arrow definition.
arrowTypeid	NUMBER NOT NULL	The identifier of the type of arrow (currently there is only one supported type of arrow).
processDefinitionId	NUMBER NOT NULL	The identifier of the process definition to which this arrow definition belongs.
label	NVARCHAR2(64) NOT NULL	The descriptive name of this arrow definition.
longRaw	BLOB NULL	The data associated with this arrow definition.
description	NVARCHAR2(256) NULL	The explanatory notes for this arrow definition.
srcActivityDefinitionId	NUMBER NOT NULL	The identifier of the node definition from which this arrow definition originates.
targetActivityDefinitionId	NUMBER NOT NULL	The identifier of the node definition into which this arrow definition terminates.
event	NVARCHAR2(64) NULL	For future expansion.
posX1	NUMBER NOT NULL	The starting point's (tail) x-coordinate of this arrow for graphical rendering.
posY1	NUMBER NOT NULL	The starting point's (tail) y-coordinate of this arrow for graphical rendering.
posX2	NUMBER NOT NULL	The ending point's (head) x-coordinate of this arrow for graphical rendering.
posY2	NUMBER NOT NULL	The ending point's (head) y-coordinate of this arrow for graphical rendering.
PRIMARY KEY(ARROWDEFINITIONID)		Each arrow definition is identified by its unique identifier.

FOREIGN KEY (processDefinitionId) references ProcessDefinition(processDefinitionId)

## ArrowInstance

This table represents an arrow instance of Interstage BPM.

Column Name	Types	Properties	Remarks
arrowInstanceid	NUMBER NOT NULL		The identifier of the arrow instance.
state	NUMBER NOT NULL	0	The state of the arrow instance.
processInstanceid	NUMBER NOT NULL		The identifier of the process instance to which this arrow instance belongs.
arrowDefinitionid	NUMBER NOT NULL		The identifier of the arrow definition from which this instance of the arrow was created.
srcActivityInstanceid	NUMBER NOT NULL		The identifier of the node instance from which this arrow instance originates.
targetActivityInstanceid	NUMBER NOT NULL		The identifier of the node instance in which this arrow instance terminates.
label	NVARCHAR2(64) NOT NULL		The descriptive name of this arrow instance.
PRIMARY KEY(ARROWINSTANCEID)			Each arrow instance is distinguished by its unique identifier.

FOREIGN KEY (processInstanceid) references ProcessInstance(processInstanceid)

## Comments

This table represents the definitions of the elements associated with each comment.

Column Name	Types	Remarks
commentid	NUMBER NOT NULL	The identifier of the comment with which the comments are being represented.
createdTime	DATE NOT NULL	The time that the comment is created.
userid	NVARCHAR2(256) NOT NULL	The identifier for user of the person entering comment.

Column Name	Types	Remarks
type	NUMBER NOT NULL	The type of comment whether it is Process Instance comment or Node Instance comment.
message	NVARCHAR2(2000) NOT NULL	The message added in the comment.
processInstancelid	NUMBER NOT NULL	The identifier of the process instance which the comment is associated with.
consumerId	NUMBER NOT NULL	The identifier of either node instance or process instance for which comment is added
deletedTimeStamp	DATE NULL	The time stamp of when the comment is deleted.
PRIMARY KEY(COMMENTID)		Each comment is distinguished by its unique commentId.

FOREIGN KEY (processInstancelid) references ProcessInstance(processInstancelid)

### DataSetDefinitionValues

This table represents the definitions of the data elements associated with each process definition.

Column Name	Types	Remarks
processDefinitionId	NUMBER NOT NULL	The identifier of the process definition with which the data elements are being represented.
data	BLOB NOT NULL	The data associated with this process definition.
PRIMARY KEY(PROCESSDEFINITIONID)		Each process definition has at most one set of data element definitions.

FOREIGN KEY (processDefinitionId) references ProcessDefinition(processDefinitionId)

### ElementLock

This table represents the lock state of process definitions and process instances.

Column Name	Types	Remarks
type	NUMBER NOT NULL	The identifier of the type of element (process definitions, process instances).
elementId	NUMBER NOT NULL	The identifier of the element.

Column Name	Types	Remarks
lockingId	NUMBER NOT NULL	The identifier of the user holding the lock for the element.
serverName	NVARCHAR2(256) NOT NULL	The name of the server holding the lock for the element.
timeLocked	DATE NOT NULL	The time when the element was locked.
PRIMARY KEY(TYPE, ELEMENTID)		Each lock is distinguished by the type of element and the identifier of the element being locked.

### GroupToGroup

This table stores associations between user groups.

Column Name	Types	Remarks
childGroup	NVARCHAR2(256) NOT NULL	The name of a group that is a child of another group.
parentGroup	NVARCHAR2(256) NOT NULL	The name of the parent group to which the child group belongs.
tenantId	NUMBER NOT NULL	Unique identifier for a tenant
PRIMARY KEY(CHILDGROUP, PARENTGROUP, TENANTID)		A group can belong to multiple other groups

FOREIGN KEY (tenantId) references Tenant(tenantId)

### Groups

This table represents user groups.

Column Name	Types	Properties	Remarks
groupName	NVARCHAR2(256) NOT NULL		The name of the user group.
groupDescription	NVARCHAR2(256) NULL		Detailed information about the user group.
readOnly	NUMBER NOT NULL		Specifies whether the user group can be removed from Interstage BPM.
tenantId	NUMBER NOT NULL		Unique identifier for a tenant
PRIMARY KEY(GROUPNAME, TENANTID)			Each group is identified by its name and tenant ID.

FOREIGN KEY (tenantId) references Tenant(tenantId)

## History

The History table maps an event in Interstage BPM. Since every action performed within Interstage BPM is an event, this table constitutes the exact state of the Interstage BPM system at all times. Each row in the History table represents an Interstage BPM event.

Column Name	Types	Properties	Remarks
historyId	NUMBER NOT NULL		Each event is assigned a unique identifier.
eventCode	NUMBER NOT NULL	0=Start (default) 1=Activate 2=MakeChoice 3=Accept 4=Decline 5=Reassign 6=Exit 7=CreateSubProcess 8=SuspendWorkItem 9=ResumeWorkItem	Each code uniquely identifies a separate event.
eventType	NVARCHAR2(256) NOT NULL		Each event type along with some relevant data (all eventTypes are defined in metamodel: WFELEMENT).
eventData	BLOB NULL		Extended data about the event (Future expansion).
createdTime	DATE NOT NULL		The time the event was created.
responsible	NVARCHAR2(256) NOT NULL		The user responsible for this event.
isHandled	NUMBER NOT NULL	0=notHandled 1=handled 2=ignored 3=Error 4=audit	The state of the event.
producerType	NUMBER NOT NULL	0=ArrowType 3=ActivityType 7=ProcessType 15=TimerType	The element type of the producer of the event.



Column Name	Types	Properties	Remarks
producerId	NUMBER NOT NULL		The identifier of the producer.
consumerType	NUMBER NOT NULL	0=ArrowType 3=ActivityType 7=ProcessType	The element type that the event is intended for.
consumerId	NUMBER NOT NULL		The identifier of the consumer of the event.
processInstanceId	NUMBER NOT NULL		The identifier of the process instance which owns the event.
serverName	NVARCHAR2(64) NULL		The name of the server for which these events are generated and handled.
applicationContainerId	NUMBER NOT NULL		Unique identifier of the application that contains this event
PRIMARY KEY(HISTORYID)			Each event is distinguished by its unique identifier.

FOREIGN KEY (processInstanceId) references ProcessInstance(processInstanceId)

### IBPMProperties

This table stores configuration parameters of the Interstage BPM Server.

Column Name	Types	Remarks
propertyKey	NVARCHAR2(2000) NOT NULL	Name of the configuration parameter.
propertyValue	NVARCHAR2(2000) NULL	Value of the configuration parameter.
tenantId	NUMBER NOT NULL	Unique identifier of a tenant
PRIMARY KEY(TENANTID,PROPERTYKEY)		Each configuration parameter is identified by its unique key, which is the parameter's name.

FOREIGN KEY (tenantId) references Tenant(tenantId)

### JsPiHandle

This table holds the process handles to be passed to the JavaScript engine.

Column Name	Types	Remarks
id	NUMBER NOT NULL	The identifier of the process instance.
longRaw	BLOB NOT NULL	The handle of the process instance.
PRIMARY KEY(ID)		Each process instance is identified by its unique identifier.

FOREIGN KEY (id) references ProcessInstance(processInstancelid)

### KeyPerfIndicators

This table stores Key Performance Indicator information.

Column Name	Types	Properties	Remarks
kpid	NUMBER NOT NULL		For any created KPI, a unique ID will be assigned in order to identify the KPI.
kpiName	NVARCHAR2(64) NOT NULL		Name for the newly created KPI.
description	NVARCHAR2(256) NOT NULL		Description of the KPI.
targetValue	NUMBER NOT NULL		The initial expected value for this KPI.
displayType	NUMBER NOT NULL	0 = Meter	Based on this value, a KPI will be displayed on a Dashboard in a particular shape. Currently, only "Meter" is supported.
kpiDisplay	NUMBER NOT NULL	0 = KPI will not appear 1 = KPI will appear	A value which decides whether to show this KPI on the Dashboard view.

---

Column Name	Types	Properties	Remarks
kpiScope	NUMBER NOT NULL	0 = System 1 = Process 2 = Activity	This column stores the value for the selected scope for this KPI. Currently, there are 3 different types of KPI: "System": All the current activities and processes in the IBPM engine will be measured. "Process": All the processes for a given process definition will be measured. "Activity": All the activities for a selected activity in the process definition will be measured.
processDefinitionId	NUMBER NOT NULL		This column stores the process definition ID used to measure the KPI for the "Process" or "Activity" scope.
activityDefinitionId	NUMBER NOT NULL		This column stores the node definition ID used to measure the KPI for the "Activity" scope.

Column Name	Types	Properties	Remarks
kpiMeasure	NUMBER NOT NULL	0 = Wait Time 1 = Amount Waiting 2 = Average Cycle Time	This column is used to store the type of measure for the selected KPI scope. The KPI value will be computed based on the measure selected for a given KPI scope. Currently, there are 3 types of KPI measures: "Wait Time": The difference between the time the process instance or activity is created and the current time. "Amount Waiting": The number of process instances or activities currently running for a selected process definition or activity. "Average Cycle Time": The average time take to close all the running process instances or activities for a selected process definition or activity.
PRIMARY KEY(KPIID)			Each KPI is identified by its unique identifier.

### MaxId

This table contains the next available identifier to be used.

Column Name	Types	Properties	Remarks
objectType	NVARCHAR2(64) NOT NULL	ActivityDefinition ActivityInstance ActivityTimer ActivityType ArrowDefinition ArrowInstance ArrowType Folder FolderFilter History ProcessDefinition ProcessInstance ProcessTimer ProcessType ServerTimer TimerInstance TimerType WorkItem	The type of element. <b>Note:</b> Only the ArrowType row is used to assign ids for all the Interstage BPM elements.

Column Name	Types	Properties	Remarks
nextId	NUMBER NOT NULL		The next identifier for the element type.
PRIMARY KEY(OBJECTTYPE)			Each type of object is distinguished by its unique type and the identifiers assigned to any two elements in Interstage BPM are always unique.

### ProcDefAccess

This table represents the internal structure of the process definition.

Column Name	Types	Remarks
id	NUMBER NOT NULL	The identifier of the process definition.
longRaw	BLOB NOT NULL	The internal structure representing the process definition.
PRIMARY KEY(ID)		Each process definition is distinguished by its identifier.

FOREIGN KEY (id) references ProcessDefinition(processDefinitionId)

### ProcessDataSet

This table stores the values of worklist UDAs.

Column Name	Types	Remarks
processInstanceId	NUMBER NOT NULL	The identifier of the process instance whose data element values are being represented.
longRaw	BLOB NULL	The data associated with this process instance.
workItemUdas	VARCHAR2(4000) NULL	Summary of all UDA allocations.
PRIMARY KEY(PROCESSINSTANCEID)		All the process data set values for a process instance are distinguished by the identifier of the process instance.

FOREIGN KEY (processInstanceId) references ProcessInstance(processInstanceId)

### ProcessDataSetDefinition

This table represents the data set definitions associated with each process definition.

Column Name	Types	Remarks
name	NVARCHAR2(64) NOT NULL	The name of the data element.
processDefinitionId	NUMBER NOT NULL	The identifier of the process definition with which it is associated.
type	NUMBER NOT NULL	The type of the data element.
attributes	NUMBER NOT NULL	The attributes of the data element.
cubeNames	NVARCHAR2(256) NULL	The cube name of the data element.
aggregates	NVARCHAR2(256) NULL	The aggregates of the data element.
id	NVARCHAR2(32) NOT NULL	The identifier of the data element.
PRIMARY KEY(NAME, PROCESSDEFINITIONID)		Each process data element is distinguished by its name and the identifier of the process definition with which it is associated.

FOREIGN KEY (processDefinitionId) references ProcessDefinition(processDefinitionId)

### ProcessDefinition

This table represents an Interstage BPM process definition.

Column Name	Types	Properties	Remarks
processDefinitionId	NUMBER NOT NULL		The identifier of the process definition.
label	NVARCHAR2(64) NOT NULL		The name of the process definition.
processTypeid	NUMBER NOT NULL		The identifier of the type of process instance that this process definition would be used to create.
templateIdentifier	NVARCHAR2(256) NULL		A user defined identifier which could be used to group families of process definitions based on business requirements.
owner	NVARCHAR2(64) NOT NULL		The name of the owner of the process definition.
title	NVARCHAR2(128) NULL		The title of the process definition, different from the name of the process definition.
description	NVARCHAR2(256) NULL		A detailed explanation of the process definition.

Column Name	Types	Properties	Remarks
state	NUMBER NOT NULL	0=Draft 1=Published 2=Private 3=Obsolete 4=Deleted	The state of the process definition.
processOwnerRole	NVARCHAR2(256) NULL		The group whose members are the valid owners of all process instances derived from this process definition.
majorVersion	NUMBER NULL	> 0	The major version of this process definition.
minorVersion	NUMBER NULL	0 (default) > 0	The minor version of this process definition.
parentVersion	NVARCHAR2(64) NULL		The version of the parent process definition, from which this process definition was copied (if from the same family).
creationTime	DATE NULL		The absolute time when the process definition was created.
organization	NVARCHAR2(256) NULL		The organization to which this process definition belongs (for future requirements).
sequenceId	NUMBER NOT NULL		Every change on this process definition is tracked using a running number in this column.
longRaw	BLOB NULL		The data associated with this process definition.
flags	NUMBER NULL		An optimization parameter recording the presence of various scripts for this process definition.
versionComment	NVARCHAR2(256) NULL		A detailed explanation of why a new version was required.

Column Name	Types	Properties	Remarks
applicationContainerId	NUMBER NOT NULL		Unique identifier for the application that contains this process definition
priority	NUMBER NOT NULL		The priority of the process definition. This will be the default priority of process instance.
PRIMARY KEY (PROCESSDEFINITIONID)			Each process definition is distinguished by its unique identifier.

FOREIGN KEY (applicationContainerId) references WorkflowApplication(applicationContainerId)

### ProcessDefinitionVersion

This table represents version information for each family of process definition.

Column Name	Types	Properties	Remarks
label	NVARCHAR2(64) NOT NULL		The name of the process definition family.
publishedProcessDefinitionId	NUMBER NOT NULL		The identifier of the published process definition in the family.
latestMajorVersion	NUMBER NOT NULL	>0	The major version of the latest process definition in the family.
latestMinorVersion	NUMBER NOT NULL	>0	The minor version of the latest process definition in the family.
latestProcessDefinitionId	NUMBER NOT NULL		The identifier of the latest process definition in the family.
applicationContainerId	NUMBER NOT NULL		Unique identifier for the application that contains this process definition
PRIMARY KEY (APPLICATIONCONTAINERID, LABEL)			Each family of process definitions is distinguished by its name and the application it belongs to.

FOREIGN KEY (applicationContainerId) references WorkflowApplication(applicationContainerId)

### ProcessInstance

This table represents an Interstage BPM process instance.



Column Name	Types	Properties	Remarks
processInstanceId	NUMBER NOT NULL		The identifier of the process instance.
processDefinitionId	NUMBER NOT NULL		The identifier of the process definition from which this process instance was derived.
parentId	NUMBER NOT NULL		For a child process instance, this is the identifier of the parent process instance.
state	NUMBER NOT NULL	0=PisOpen 1=PisNotRunning 2=PisNotStarted 3=Created 4=Initialized 5=Dormant 6=Locked 7=PisRunning 8=PisClosed 9=PisCompleted 10=PisError 11=PisAborted 12=PisSuspend	The state of the process instance.
initiator	NVARCHAR2(256) NOT NULL		The person responsible for starting the process instance.
priority	NUMBER NOT NULL	0=PRIORITY_LOW 8=PRIORITY_MEDIUM 16=PRIORITY_HIGH	The priority of the process instance.
createdTime	DATE NOT NULL		The absolute time when the process instance was first created.
label	NVARCHAR2(64) NULL		The name of the process instance.
title	NVARCHAR2(128) NULL		The title provided to the process instance, different from the name of the process instance.
description	NVARCHAR2(256) NULL		A detailed explanation of the process instance.

Column Name	Types	Properties	Remarks
attachments	BLOB NULL		The attachment data associated with this process instance.
closedTime			The absolute time when the process instance was closed.
sequenceId	NUMBER NULL		Every change on this process instance is tracked using a running number in this column.
parentProcessURI	NVARCHAR2(2000) NULL		URI of the process definition this process instance is derived from.
DueDateExpireTime	DATE NOT NULL		Duedate for this Process Instance
lastUpdatedTime	DATE NOT NULL		The absolute time when this process instance was last updated/created.
PRIMARY KEY (PROCESSINSTANCEID)			Each process instance is distinguished by its unique identifier.

FOREIGN KEY (processDefinitionId) references ProcessDefinition(processDefinitionId)

### ProcessInstanceOwner

This table represents the owner(s) for every Interstage BPM process instance.

Column Name	Types	Remarks
owner	NVARCHAR2(256) NOT NULL	The name of the owner of the process instance.
processInstanceid	NUMBER NOT NULL	The identifier of the process instance.
PRIMARY KEY(OWNER, PROCESSINSTANCEID)		There can be more than one owner for a given process instance.

FOREIGN KEY (processInstanceid) references ProcessInstance(processInstanceid)

### ProcessTimerDefinition

This table represents the timer definitions associated with a process definition.

Column Name	Types	Remarks
timerId	NUMBER NOT NULL	The identifier of the process timer definition.
processDefinitionId	NUMBER NOT NULL	The process definition with which this timer is associated.
label	NVARCHAR2(64) NULL	The name of the timer.
data	BLOB NULL	The data associated with this timer.
IsDueDate	NUMBER NULL	Specifies if this is a due date timer
PRIMARY KEY(TIMERID)		Each process timer definition is distinguished by its unique identifier.

FOREIGN KEY (processDefinitionId) references ProcessDefinition(processDefinitionId)

### ProcInstAccess

This table represents the internal structure of the process instance.

Column Name	Types	Remarks
id	NUMBER NOT NULL	The identifier of the process instance.
longRaw	BLOB NOT NULL	The internal structure representing the process instance.
PRIMARY KEY(ID)		Each process instance is distinguished by its identifier.

FOREIGN KEY (id) references ProcessInstance(processInstancelId)

### Responses

Currently, this table is NOT used. This table may be used in the future to implement the different kinds of actions to be performed on a response.

Column Name	Types	Remarks
responselId	NUMBER NOT NULL	Unique ID for newly created responses.
thresholdId	NUMBER NOT NULL	Unique ID for a threshold for which this response is generated.
actionType	NUMBER NOT NULL	Different types of action that will be performed for this response. Currently not supported.
actionName	NVARCHAR2(64) NOT NULL	Name of the action to be performed for this response. Currently not supported.

Column Name	Types	Remarks
actionDetails	NCLOB NOT NULL	Description of the action to be performed as part of the response.
PRIMARY KEY(RESPONSEID)		Each response is distinguished by its identifier.

FOREIGN KEY (thresholdId) references Thresholds(thresholdId)

## Tenant

This table stores information related to tenants

Column Name	Types	Remarks
tenantId	NUMBER NOT NULL	Unique ID for newly created tenant.
name	NVARCHAR2(64) NOT NULL	Name for newly created tenant.
description	NVARCHAR2(256) NULL	Description of the tenant
state	NUMBER NOT NULL	State of the tenant: 2: activated 1: deactivated
longRaw	BLOB NULL	
PRIMARY KEY(TENANTID)		Each tenant is distinguished by its identifier.

## Thresholds

This table stores information on the threshold which apply to a particular Key Performance Indicator.

Column Name	Types	Properties	Remarks
thresholdId	NUMBER NOT NULL		Unique ID for a newly created threshold.
thresholdType	NUMBER NOT NULL	0 = Initial Threshold 1 = Critical Threshold	Stores the value for the type of threshold for a given KPI. Currently, there are 2 types of threshold: "Initial Threshold": Used to trigger warning messages when the system reaches this threshold level. "Critical Threshold": Used to perform some action to make the system healthier again when the system reaches this threshold level.

Column Name	Types	Properties	Remarks
description	NVARCHAR2(256) NOT NULL		Detailed information about this threshold.
kpild	NUMBER NOT NULL		Unique ID of the KPI for which this threshold has been created.
thresholdValue	NUMBER NOT NULL		Actual value of the threshold, which defines the initial and critical regions for a given KPI.
processDefinitionId	NUMBER NOT NULL		The ID of the process definition that will be instantiated when the KPI reaches one of the thresholds.
alarmFrequency	NUMBER NOT NULL		Stores, in minutes, the value how frequently the process needs to be instantiated to take some action as it reaches the thresholds.
alarmTime	NUMBER NOT NULL		The actual time when this process will be instantiated to take some kind of action once it reaches the thresholds. Currently not used.
PRIMARY KEY(THRESHOLDID)			Each threshold is distinguished by its identifier.

FOREIGN KEY (kpild) references KeyPerfIndicators(kpild)

### TimerInstance

This table represents an instance of a timer.

Column Name	Types	Properties	Remarks
timerInstanceid	NUMBER NOT NULL		The identifier of the timer instance.
timerDefinitionId	NUMBER NOT NULL		The identifier of the timer definition from which this instance was created.
realTimeInMillis	NVARCHAR2(64) NOT NULL		The absolute time when this timer will expire.

Column Name	Types	Properties	Remarks
deltaTimeInMillis	NVARCHAR2(64) NOT NULL		The relative time compared to the time of creation of this timer instance, after which this timer will expire.
timerTypeId	NUMBER NOT NULL		The identifier of the type of this timer {node, process instance, server}.
eventType	NVARCHAR2(256) NOT NULL		The type of the event (TimerEvent).
eventCode	NUMBER NULL		The identifier of the event.
eventData	BLOB NULL		The data associated with this timer instance (for future expansion).
timerTimeStamp	DATE NULL		The time when this instance was created.
responsible	NVARCHAR2(256) NOT NULL		The person responsible for this timer.
isHandled	NUMBER NULL	0=notHandled 1=handled 2=ignored 3=Ehterror 4=audit 5=suspended 6=Analytics	The state of this timer.
producerType	NUMBER NULL	0=ArrowType 3=ActivityType 7=ProcessType	The Interstage BPM element type which generated this instance.
producerId	NUMBER NOT NULL		The identifier of the element that created this timer instance.
consumerType	NUMBER NULL	0=ArrowType 3=ActivityType 7=ProcessType	The Interstage BPM element type which will act upon this instance.
consumerId	NUMBER NOT NULL		The identifier of the element that will act upon this timer instance.

Column Name	Types	Properties	Remarks
processInstanceid	NUMBER NOT NULL		The identifier of the process instance with which this timer is associated.
serverName	NVARCHAR2(64) NULL		The name of the server for which this timer instance is generated and handled.
PRIMARY KEY(TIMERINSTANCEID)			Each timer instance is distinguished by its unique identifier.

FOREIGN KEY (processInstanceid) references ProcessInstance(processInstanceid)

### TriggerDefinition

This table stores trigger definition information.

Column Name	Types	Properties	Remarks
id	NUMBER NOT NULL		Trigger definition ID
typeid	NUMBER NOT NULL	0 = Start Process 1 = Make Choice	Trigger type ID
processDefinitionId	NUMBER NOT NULL		The ID of the process definition containing the trigger.
activityDefinitionId	NUMBER NOT NULL		The ID of the node definition containing the trigger. The value is -1 for process level triggers (e.g. start process trigger).
state	NUMBER NOT NULL	0 = Default 1 = Active 2 = Inactive	State of the trigger.
label	NVARCHAR2(64) NOT NULL		Name of the trigger
description	NVARCHAR2(256) NULL		Description text of the trigger
scripts	NCLOB NULL		Trigger script in XML format

Column Name	Types	Properties	Remarks
hint	NVARCHAR2(2000) NULL		Trigger hint stored in XML format. The trigger hint can be set in the trigger event panel. It contains the following items: <ul style="list-style-type: none"> <li>- Event Type</li> <li>- Event Schema URL</li> <li>- Event Filter Expression</li> </ul>
PRIMARY KEY(ID)			Each trigger is distinguished by its unique identifier.

FOREIGN KEY (processDefinitionId) references ProcessDefinition(processDefinitionId)

### UserProfile

This table represents user preferences.

Column Name	Types	Properties	Remarks
userProfileId	NVARCHAR2(256) NOT NULL		The identifier of the user.
userProfile	BLOB NULL		The data associated with this user.
tenantId	NUMBER NOT NULL		Unique identifier for a tenant
PRIMARY KEY(USERPROFILEID, TENANTID)			Each Interstage BPM user's preference is distinguished by the user's unique identifier and tenant ID.

FOREIGN KEY (tenantId) references Tenant(tenantId)

### UserToGroup

This table stores associations between users and user groups.

Column Name	Types	Remarks
userId	NVARCHAR2(256) NOT NULL	The identifier of the user.
groupName	NVARCHAR2(256) NOT NULL	The name of the group to which the user belongs.
tenantId	NUMBER NOT NULL	Unique identifier for a tenant



Column Name	Types	Remarks
PRIMARY KEY(USERID,GROUPNAME, TENANTID)		A user can belong to multiple groups.

FOREIGN KEY (tenantId) references Tenant(tenantId)

## Users

This table stores user accounts.

Column Name	Types	Properties	Remarks
userId	NVARCHAR2(256) NOT NULL		The identifier of the user.
password	NVARCHAR2(2000) NOT NULL		The password of the user.
readOnly	NUMBER NOT NULL		Specifies whether the user can be removed from Interstage BPM.
tenantId	NUMBER NOT NULL		Unique identifier for a tenant
PRIMARY KEY(USERID, TENANTID)			Each user is distinguished by the user's identifier and tenant ID.

FOREIGN KEY (tenantId) references Tenant(tenantId)

## WLUdaValues

This table stores the values of worklist UDAs. The table is used for filtering and sorting.

Column Name	Types	Properties	Remarks
processInstanceId	NUMBER NOT NULL		The identifier of the process instance.
udaName	NVARCHAR2(64) NOT NULL		The name of the UDA.
udaType	NUMBER NOT NULL	0 = BOOLEAN 1 = STRING 2 = INTEGER 3 = FLOAT 4 = LONG 8 = BIGDECIMAL 9 = DATE 11=XML	The data type of the UDA.

Column Name	Types	Properties	Remarks
bigdecimalValue	NVARCHAR2(256) NULL		The value of the UDA if the data type is BIGDECIMAL.
stringValue	NVARCHAR2(256) NULL		The value of the UDA if the data type is STRING.
intValue	NUMBER NULL		The value of the UDA if the data type is INTEGER.
longValue	NUMBER NULL		The value of the UDA if the data type is LONG.
dateValue	NUMBER NULL		The value of the UDA if the data type is DATE.
booleanValue	NUMBER NULL		The value of the UDA if the data type is BOOLEAN.
floatValue	NUMBER NULL		The value of the UDA if the data type is FLOAT.
udald	NVARCHAR2(32) NOT NULL		The identifier of the UDA.
xmlValue	XMLType NULL		The XML value of the UDA if it is a XML/custom UDA.  To optimize the performance of search queries on XML/custom UDAs, this column needs to be indexed in accordance with the data stored and the queries that need to be executed. You need to create these indexes; the Interstage BPM database scripts do not do this.  For databases that do not support XMLType column, this column is not required.
PRIMARY KEY (PROCESSINSTANCEID, UDANAME)			Each element is distinguished by its unique process instance identifier and name.

FOREIGN KEY (processInstancecd) references ProcessInstance(processInstancecd)

## WorkItem

While a node instance represents a single activity in a process, a work item is a user's representation of the node instance. Every user that is a valid user for a given node instance, gets his/her own private copy of the node instance, which is the work item.

Column Name	Types	Remarks
workItemId	NUMBER NOT NULL	The identifier of the work item.
activityInstanceid	NUMBER NOT NULL	The identifier of the node instance of which this work item is a user copy.
responsible	NVARCHAR2(256) NOT NULL	The user to which this work item belongs.
state	NUMBER NOT NULL	The state of the work item.
createdTime	DATE NOT NULL	The time when the work item was created.
priority	NUMBER NOT NULL	The priority of the work item.
label	NVARCHAR2(64) NOT NULL	The descriptive name of the work item.
processInstanceid	NUMBER NOT NULL	The identifier of the process to which the work item belongs.
choice	NVARCHAR2(64) NULL	In case of Voting User Task Nodes, this represents the choice made on the work item.
responsibleUser	NVARCHAR2(256) NULL	The user responsible for accept or start subprocess operations on the work item.
applicationContainerid	NUMBER NOT NULL	Unique identifier of the application that contains this work item
PRIMARY KEY(WORKITEMID)		Each work item is distinguished by the identifier of the work item.

FOREIGN KEY (processInstanceid) references ProcessInstance(processInstanceid)

## WorkItemOperations

This table represents data for decline and read operations on a group work item.

Column Name	Types	Properties	Remarks
activityInstanceid	NUMBER NOT NULL		The identifier of the node instance of which this group work item is a user copy.

Column Name	Types	Properties	Remarks
operator	NVARCHAR2(256) NOT NULL		The name of the user who performed the decline or read operation.
operation	NUMBER NOT NULL	1=read 2=declined	The operation performed on the group work item.
PRIMARY KEY (ACTIVITYINSTANCEID, OPERATOR)			Each group work item is distinguished by the unique identifier of the node instance it belongs to and by its operator.

FOREIGN KEY (activityInstanceid) references ActivityInstance(activityInstanceid)

### WorkflowApplication

This table stores the state of Workflow Applications.

Column Name	Types	Properties	Remarks
applicationId	NVARCHAR2(256) NOT NULL		The name of the Workflow Application.
applicationState	NUMBER NOT NULL		The state of the Workflow Application.
tenantId	NUMBER NOT NULL		Unique ID for newly created tenant.
applicationContainerId	NUMBER NOT NULL		The unique identifier for the application
ownerGroup	NVARCHAR2(200) NOT NULL		Name of the group that has access to this application
PRIMARY KEY (APPLICATIONCONTAINERID)			Each Workflow Application is identified by this unique identifier.

FOREIGN KEY (tenantId) references Tenant(tenantId)

## Appendix B: Message List of Interstage BPM Server

This section explains the description and the action method of messages outputting to `Status.log`.

### Message ID: 00001

Name	Details
Error Level	INFO
Message Text	Interstage BPM server has started.
Description	Interstage BPM server started successfully.

### Message ID: 00002

Name	Details
Error Level	ERROR
Message Text	Interstage BPM server startup failed.
Description	Interstage BPM server startup failed.
System processing	Stopped Interstage BPM server abnormally, or tried to restart Interstage BPM server by Self-Healing. Restart is retried until the problem is resolved.
Action Method	<p>If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause. Then, restart the Interstage BPM server if it stops.</p> <p>If the problem is not resolved, collect a set of Interstage BPM server logs and contact Fujitsu technical support.</p>

### Message ID: 00003

Name	Details
Error Level	ERROR
Message Text	Error detected during system health check.
Description	Error was detected during system health check.
System processing	Tries to restart Interstage BPM server by Self-Healing. Restart is retried until the problem is resolved.
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause.

### Message ID: 00011

Name	Details
Error Level	ERROR

Name	Details
Message Text	Failed to initialize tenant.: TENANT=%s %s: Tenant name
Description	Tenant initialization failed.
System processing	Tries to reinitialize the tenant by Self-Healing. The initialization is retried until the problem is resolved.
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause. If the problem is not resolved, collect a set of Interstage BPM server logs and contact Fujitsu technical support.

**Message ID: 00012**

Name	Details
Error Level	INFO
Message Text	Tenant initialized.: TENANT=%s %s: Tenant name
Description	Tenant initialization completed.

**Message ID: 00013**

Name	Details
Error Level	ERROR
Message Text	Error detected during tenant health check.: TENANT=%s %s: Tenant name
Description	Error was detected during tenant health check.
System processing	Tries to reinitialize the tenant by Self-Healing. The initialization is retried until the problem is resolved.
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause. If the problem is not resolved, collect a set of Interstage BPM server logs and contact Fujitsu technical support.

**Message ID: 00051**

Name	Details
Error Level	ERROR

Name	Details
Message Text	Failed to initialize workflow application.: TENANT=%s1 APPLICATION=%s2 %s1: Tenant name %s2: Application ID
Description	Workflow application initialization failed.
System processing	Changes the workflow application state to offline.
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause. Then, start the workflow application. If the problem is not resolved, collect a set of Interstage BPM server logs and contact Fujitsu technical support.

**Message ID: 00101**

Name	Details
Error Level	ERROR
Message Text	Interstage BPM server internal error occurred.
Description	Internal error occurred.
System processing	Processing is aborted.
Action Method	Collect a set of Interstage BPM server logs and contact Fujitsu technical support.

**Message ID: 00201**

Name	Details
Error Level	ERROR
Message Text	Database access failed.: CODE=%s1 MSG=%s2 %s1: Error code <ul style="list-style-type: none"> <li>• Error code type <ul style="list-style-type: none"> <li>• 1: Connection failure</li> <li>• 0: Others</li> </ul> </li> </ul> %s2: Detailed information
Description	Error occurred when database access.
System processing	Processing is aborted.

Name	Details
Action Method	If error code is equal to "1", confirm whether database is running correctly and is accessible. If database is not running, then start the database. Also confirm whether there is any process instance in "Error" state or not by using Interstage BPM console. If any process instance is in "Error" state, then edit and correct the process instance so that it returns to "Running" state.

**Message ID: 00301**

Name	Details
Error Level	ERROR
Message Text	Failed to send JMS message.: DEST=%s %s: JNDI name of destination
Description	Sending JMS (Java Message Service) message failed.
System processing	Aborts to send JMS message.
Action Method	Confirm whether the JMS queue or topic for the destination is accessible or not.

**Message ID: 00302**

Name	Details
Error Level	WARNING
Message Text	Failed to send JMS message.: DEST=%s %s: JNDI name of destination
Description	Sending JMS (Java Message Service) message failed.
System processing	Interstage BPM server recovers the processing.
Action Method	If this message is repeatedly output, confirm whether the JMS queue or topic for the destination is accessible or not.

**Message ID: 00401**

Name	Details
Error Level	ERROR
Message Text	Directory server access failed.: CODE=%s %s: Error code <ul style="list-style-type: none"> <li>• Error code type: <ul style="list-style-type: none"> <li>• 1: Connection failure</li> <li>• 0: Others</li> </ul> </li> </ul>
Description	Error occurred when directory server access.



Name	Details
System processing	Processing is aborted.
Action Method	If error code is equal to "1", confirm whether directory server is running correctly and is accessible. If directory server is not running, then start the directory server.

**Message ID: 00501**

Name	Details
Error Level	ERROR
Message Text	Failed to initialize multicast socket.
Description	Initialization of multicast socket failed.
System processing	Processing is aborted.
Action Method	<p>Confirm whether system configuration parameter <code>ClusterMultiCastPort</code> and <code>ClusterMulticastTopic</code> is correct or not. To confirm whether multicast connection is available, you can use the following commands located in <code>&lt;engine directory&gt;/server/deployment/bin</code>.</p> <p><code>startMulticastReceiver</code> and <code>startMulticastSender</code> command (Regarding the commands, refer to <i>Server and Console Installation Guide</i>)</p> <p>Also confirm whether all the clustered nodes belong to the same sub-network and the multicast port is not blocked by firewall.</p>

**Message ID: 00551**

Name	Details
Error Level	ERROR
Message Text	Failed to send Email.
Description	Sending Email failed.
System processing	Aborts to send Email message.
Action Method	Confirm whether system configuration parameter related to SMTP server ( <code>SMTPServerHost</code> , <code>SMTPServerPort</code> , <code>SMTPUserName</code> , <code>SMTPPassword</code> ) is correct or not. Also confirm whether SMTP server is accessible or not.

**Message ID: 00552**

Name	Details
Error Level	ERROR

Name	Details
Message Text	Failed to receive Email.: TENANT=%s %s: Tenant name
Description	Receiving Email failed.
System processing	Aborts to receive Email message.
Action Method	Confirm whether tenant configuration parameter related to POP3 server (EmailListenerServerHost, EmailListenerPOPPort, EmailListenerUserName, EmailListenerPassword) is correct or not. Also confirm whether POP3 server is accessible or not.

**Message ID: 00601**

Name	Details
Error Level	WARNING
Message Text	Failed to schedule timer.
Description	Error occurred during timer schedule.
System processing	Retries to schedule the timer.
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause.

**Message ID: 00602**

Name	Details
Error Level	WARNING
Message Text	Failed to recover unhandled events.
Description	Error occurred during periodic recovery for unhandled events.
System processing	The events will be handled during the next periodic recovery. This recovery is executed every 15 minutes.
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause.

**Message ID: 00603**

Name	Details
Error Level	WARNING
Message Text	Error occurred during fail over.
Description	Error occurred during the processing of automated failover on load-balancing system.

---

Name	Details
System processing	Aborts the processing of automated failover. Other server node will handle the event, or this server node will retry after the time specified in system configuration parameter <code>ClusterHeartBeatInterval</code> .
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause.

**Message ID: 00604**

Name	Details
Error Level	WARNING
Message Text	Failed to update search index for full text search.
Description	Updating search index for full text search failed.
System processing	Retries after the time specified in system configuration parameter <code>SearchIndexUpdateInterval</code> .
Action Method	If some messages are output just before this message, resolve the problem. If any message is not output, refer to the <code>Error.log</code> of Interstage BPM server and eliminate the error cause.

## Appendix C: Troubleshooting

### C.1 Log File Information

For details about Log Files, refer the **Configuring Log Files** topic in the *Interstage BPM Server Administration Guide*.

### C.2 Resolving Specific Error Situations

#### C.2.1 Interstage BPM Server Fails to Start

Check the `Error.log` file in the `<engine directory>/server/instance/default/logs` directory.

Look for	What to do
<pre>DbService : setConnection: Connection to database server failed. Is the database server running and reachable through the network? {ORA-01089: immediate shutdown in progress - no operations are permitted.</pre>	<p>Check if the database is running. Also check that you can access the database from the machine where the Interstage BPM Server is setup in case the database is running on a different machine. You can use <code>telnet &lt;Database Server Hostname&gt; &lt;Port&gt;</code> from the server host machine to check that the connection to the database host/port can be established.</p>
<pre>LdapBroker : getContext: Could not create the directory services. {[LDAP: error code 49 - Invalid Credentials]} LdapBroker : getGroupMembersByDN: Could not retrieve the user groups. {Could not create the directory services. {[LDAP: error code 49 - Invalid Credentials]}}</pre> <pre>LdapBroker : Could not retrieve the user groups. {Could not create the directory services. {[LDAP: error code 49 - Invalid Credentials]}}</pre>	<p>Ensure that the user name/password as specified in the <code>LDAPAccessUserID / LDAPAccessUserPassword</code> parameters of the Interstage BPM Server are correct and you can login to your Directory Server using the above user name/password.</p>
<pre>getContext: Could not create the directory services. LdapBroker : getGroupMembersByDN: Could not retrieve the user groups. {Could not create the directory services. LdapBroker : Could not retrieve the user groups. {Could not create the directory services.</pre>	<p>Ensure that the LDAP Server is running on the port as specified in the <code>LDAPServer</code> parameter of the Interstage BPM Server. You can use <code>telnet ldapServerHostName port</code> from the server host machine to check that the connection to the host/port can be established.</p>

Look for	What to do
<pre>JDBC module: ERROR: Failed to load module for driver [&lt;jdbc_module_name&gt;]" WARN : Error looking up datasources/iFlowDS, service service jboss.naming.context.java.jboss.datasources.iFlowDS is not started</pre>	<p>You need to manually add the JDBC module with the same name as specified in the <code>setup.config</code> file.</p> <p>Refer to <i>Setting-up the JDBC Module</i> for more information about how to manually configure the JDBC module.</p> <p>Restart the JBoss EAP 6 application server for the changes to take effect.</p>

## C.2.2 Error in Trace.log

Check the `Trace.log` in the `<engine_directory>/server/instance/default/logs` directory.

Look for	What to do
<pre>getGroupMembersByDN: Could not retrieve the user groups. {[LDAP: error code 32 - No Such Object]}</pre>	<p>Possible cause of this error: A user has been deleted from the Directory Server (LDAP Server) but the reference of it is still there in one of the groups.</p>

## C.2.3 Timeout During JavaScript Execution

When executing large JavaScripts, the value for transaction timeout may be insufficient for the following application server:

- WebLogic Application Server

Due to this setting, script execution may fail with a "transaction timeout".

Your application server administrator can increase the transaction timeout depending on your usage requirements, for example, to 200 seconds. This setting can be changed in the following location:

- WebLogic Application Server: `<jta>/<timeout-seconds>tag`  
in `<MW_HOME>/user_projects/domains/<Domain Name>/config/config.xml` file.  
For Example: `<jta><timeout-seconds>200</timeout-seconds></jta>`

## C.2.4 Failure in Writing to an Oracle Database

When the updating of an Oracle database table fails, for example, when you try to archive a process instance, check the Oracle alert log file located in the `<Oracle Installation Dir>/admin/<DB instance name>/bdump` directory, for example:

```
C:\ProgramFiles\Oracle\admin\orcl\bdump>alert_orcl.log
```

The following error may be observed:

```
{Database add/create request failed. {ORA-08103: object no longer exists}}
```

This failure may be due to the fact that the Datafile size reached the file size limit on the hard disk of the database server.

The system administrator of the database server needs to increase the file size on the database server hard disk.

### C.3 Errors during Setup of the Interstage BPM Server

If you have trouble executing the setup script, you can check the cause of error by viewing `deployment.log`. The deployment log is created at the following location:

On Windows: `<engine directory>\server\deployment\logs\deployment.log`

On Unix: `<engine directory>/server/deployment/logs/deployment.log`

The following tables explain the possible causes of errors during server startup and the appropriate action(s) to take:

#### Setup of IBPM fails when trying to setup a new build

I	Cause	The build directory of the existing installed build was deleted, without un-installing the build or un-installation failed.
	Action (For Windows)	<p>Manually delete the earlier build's registry-entry, as follows:</p> <ol style="list-style-type: none"> <li>1. Go to <b>Start &gt; Run</b>, type <code>regedit</code>, click <b>OK</b>.</li> <li>2. In the <b>Registry Editor</b> screen, go to below path. <ul style="list-style-type: none"> <li>• 32bit OS: <b>HKEY_LOCAL_MACHINE &gt; SOFTWARE &gt; Fujitsu &gt; Install &gt; Interstage BPM Server</b></li> <li>• 64bit OS: <b>HKEY_LOCAL_MACHINE &gt; Wow6432Node &gt; SOFTWARE &gt; Fujitsu &gt; Install &gt; Interstage BPM Server</b></li> </ul> </li> <li>3. Delete the <b>Interstage BPM Server</b> registry entry under <b>Install</b> key.</li> <li>4. Go to below path. <ul style="list-style-type: none"> <li>• 32bit OS: <b>HKEY_LOCAL_MACHINE &gt; SOFTWARE &gt; Microsoft &gt; Windows &gt; CurrentVersion &gt; Fujitsu &gt; Interstage BPM Server</b></li> <li>• 64bit OS: <b>HKEY_LOCAL_MACHINE &gt; Wow6432Node &gt; SOFTWARE &gt; Microsoft &gt; Windows &gt; CurrentVersion &gt; Fujitsu &gt; Interstage BPM Server</b></li> </ul> </li> <li>5. Delete the <b>Interstage BPM Server</b> registry entry under <b>Fujitsu</b> key. This will ensure complete deletion of the registry entry for Windows platform.</li> <li>6. Go to below path. <ul style="list-style-type: none"> <li>• 32bit OS: <b>HKEY_LOCAL_MACHINE &gt; SOFTWARE &gt; Microsoft &gt; Windows &gt; CurrentVersion &gt; Uninstall &gt; DFC70E37-68E7-469C-A253-9ED7BB3BCD23</b></li> <li>• 64bit OS: <b>HKEY_LOCAL_MACHINE &gt; Wow6432Node &gt; SOFTWARE &gt; Microsoft &gt; Windows &gt; CurrentVersion &gt; Uninstall &gt; DFC70E37-68E7-469C-A253-9ED7BB3BCD23</b></li> </ul> </li> <li>7. Delete the <b>Interstage Business Process Manager xx.x</b> registry entry under <b>Uninstall</b> key. This will ensure complete deletion of the registry entry for Windows platform.</li> </ol>

	Action (For Solaris)	<p>Manually delete the Solaris package information, as follows:</p> <ol style="list-style-type: none"> <li>1. Open the Command Prompt window and run the command <code>pkginfo -l FJSVibpm</code> from any location of the Solaris machine, to check if the Solaris package information still exists.</li> <li>2. If the Solaris package information is displayed, then delete the package information using the following steps: <ol style="list-style-type: none"> <li>a. Create a file named <b>ibpm.uninst</b> under the <b>/tmp</b> location.</li> <li>b. Run the command <code>pkgrm FJSVibpm</code> to delete the Solaris package.</li> <li>c. Run the command <code>pkginfo -l FJSVibpm</code> once more, to ensure that the Solaris package has been deleted successfully. If no information is displayed, then it confirms that the Solaris package has been completely deleted from the Solaris platform.</li> </ol> </li> </ol>
	Action (For Linux)	<p>Manually delete the RPM package information, as follows:</p> <ol style="list-style-type: none"> <li>1. Open the Command Prompt window and run the command <code>rpm -qi FJSVibpm</code> from any location of the Linux machine, to check if the RPM package still exists.</li> <li>2. If the RPM package information is displayed, then delete the package information using the command <code>rpm -e FJSVibpm</code>.</li> <li>3. Run the command <code>rpm -qi FJSVibpm</code> once more to ensure that the RPM package has been deleted successfully. If no information is displayed, then it confirms that the RPM package has been completely deleted from the Linux platform.</li> </ol>

### An error occurred during Interstage BPM database creation/update

I	Cause	<p>One of the following values provided during setup were wrong:</p> <ul style="list-style-type: none"> <li>• Database Administrator user name</li> <li>• Database Administrator password</li> </ul>
	Action	<p>Restore the database from the backup of the database made before setting up Interstage BPM.</p> <p>Update the <code>setup.config</code> file and run the setup script again.</p>
II	Cause	<p>One of the following values provided during setup were wrong:</p> <ul style="list-style-type: none"> <li>• Host name of the database server</li> <li>• Database SID (database instance name)</li> <li>• Database port</li> </ul>
	Action	<p>Restore the database from the backup of the database made before setting up Interstage BPM.</p> <p>Update the <code>setup.config</code> file and run the setup script again.</p>
III	Cause	<p>Database server is not running.</p>
	Action	<p>Start the database server.</p> <p>Update the <code>setup.config</code> file and run the setup script again.</p>

### Errors caused by JDBC Connection

I	Cause	The SQLRecoverableException occurs. This is caused by Oacle Bug 6485149.
	Action	Please contact Oracle Technical Support team for further instructions.

### Errors caused by JDK Version

I	Cause	The version of JDK specified during setup does not match the version required for this application server.
	Action	You need to update <code>JAVA_HOME</code> variable of system environment and run the setup script again. Please update <code>JAVA_HOME</code> with <code>setIBPMJava.bat/setIBPMJava.sh</code> that exists in <code>&lt;engine directory&gt;/server/deployment/bin</code> when the error happens when the script of Interstage BPM is executed after completing the setup.

### Problems related to LDAP Server Access

I	Cause	Even if LDAP server is up, the error message <code>Unable to connect to specified directory server</code> appears. This happens when Super user or Tenant Administrator is not registered in LDAP server.
	Action	Please register the Super user and Tenant Administrator in the LDAP server before executing the setup script.

### Errors Pertaining to Active Directory

I	Cause	The Active Directory Server is remote and Active Directory is not running.
	Action	Start the Active Directory Server, and then run the setup script again.
II	Cause	Interstage BPM cannot connect to the Active Directory Server because one of the following values provided during setup was wrong: <ul style="list-style-type: none"> <li>Active Directory Key</li> <li>Active Directory Organizational Unit</li> </ul>
	Action	Update the <code>setup.config</code> file and run the setup script again.

### Errors Pertaining to Oracle Directory Server Enterprise Edition

I	Cause	Interstage BPM cannot connect to the LDAP Server because one of the following values provided during setup was wrong: <ul style="list-style-type: none"> <li>LDAP Key</li> <li>LDAP Organizational Unit</li> </ul>
	Action	Update the <code>setup.config</code> file and run the setup script again.



## Errors Pertaining to Un-setup

For WebLogic Application Server:

I	Cause	Interstage BPM Un-setup failed.
	Action	<p>If un-setup fails, follow these steps to remove the resources:</p> <ol style="list-style-type: none"> <li>1. Login to WebLogic Admin Console and navigate to <b>Domain Structure</b> &gt; &lt;Domain Name&gt;.</li> <li>2. Check if the following resources exists and if exists then delete them. <ul style="list-style-type: none"> <li>• Navigate to <b>Deployments</b> and delete <code>InterstageBPMServer</code> and <code>InterstageBPMConsole</code> applications, if exists.</li> <li>• Navigate to <b>Environment</b>&gt; <b>Servers</b> and delete <code>IBPMServer</code>, if exists.</li> <li>• Navigate to <b>Services</b> &gt; <b>Messaging</b>&gt; <b>JMS Modules</b> and delete <code>InterstageBPM-JMSSystemResource.IBPMServer</code>, if exists.</li> <li>• Navigate to <b>Services</b> &gt; <b>Messaging</b>&gt; <b>JMS Servers</b> and delete <code>InterstageBPMJMSServer.IBPMServer</code>, if exists.</li> <li>• Navigate to <b>Services</b> &gt; <b>Data Sources</b> and delete <code>iflow.iFlowDS.IBPMServer</code>, if exists.</li> <li>• Navigate to <b>Services</b> &gt; <b>XML Registries</b> and delete <code>ibpm_crimson_registry</code>, <code>ibpm_sun_xerces_registry</code>, and <code>ibpm_xerces_registry</code> if exists.</li> </ul> </li> <li>3. Delete the following directories manually from the file system: <ul style="list-style-type: none"> <li>• <code>InterstageBPMServer</code> directory located at <code>&lt;Weblogic Home&gt;/Middleware/user_projects/domains/&lt;Domain Name&gt;/servers/AdminServer/upload/</code> directory.</li> <li>• <code>IBPMServer</code> directory at <code>&lt;Weblogic Home&gt;/Middleware/user_projects/domains/&lt;Domain Name&gt;/servers/</code> directory.</li> </ul> </li> </ol>

## C.4 Errors Related to Interstage BPM Database Creation/Update

### Errors Pertaining to the Database

I	Cause	The Database Server is not running.
	Action	Start the Database Server, and then start the Interstage BPM Server.

### Errors Pertaining to a Hostname Change

I	Cause	You changed the hostname of the computer where Interstage BPM Server has been setup. As the hostname occurs in the names and values of various configuration parameters of the Interstage BPM Server, the server cannot access its configuration settings.
---	-------	--

Action	<p>In the <code>IBPMProperties</code> table of the Interstage BPM database, make the following changes:</p> <ul style="list-style-type: none"> <li>In the <code>PROPERTYKEY</code> column, update any parameter names that have the hostname in the suffix. These parameters have the format <code>&lt;PARAMETER_NAME&gt;.&lt;HOSTNAME&gt;</code> or <code>&lt;PARAMETER_NAME&gt;.&lt;HOSTNAME&gt;.&lt;SERVERNAME&gt;</code>.</li> <li>In the <code>PROPERTYVALUE</code> column, update any parameter values containing the hostname.</li> </ul> <p>To update the <code>IBPMProperties</code> table, use the appropriate database commands or a database client software.</p>
--------	---

## C.5 Contacting Your Local Fujitsu Support Organization

If you are unable to troubleshoot your problem:

1. Replicate the actions that caused the error.
2. Contact your local Fujitsu Support organization and provide the following information:

### General Information

- Operating System
- Directory Service (type and version)
- Database server (type and version)
- JDK version
- Application server (type and version)
- Interstage BPM edition, version and build number
- Major problem area
- Priority of the issue
- Environment in which the problem occurs

### Configuration Information

- The configuration file that you exported from the Interstage BPM Server

### Log Files

- All log files from `<engine directory>/server/instance/default/logs`
- Setup log file from `<engine directory>/server/deployment/logs/deployment.log`
- When using WebLogic: All log files from `<MW_HOME>/user_projects/domain/<Your Domain>/servers/AdminServer/logs`

### OS System Logs

- The Windows event log that you obtain using the Windows Event Viewer
- Linux system logs stored in `/var/log/messages`
- Solaris system logs stored in `/var/adm/messages`

### Problem Description

- Description of the steps you performed before the problem occurred
- Frequency with which the problem occurs

### Problem Details

- The application program and its source code that caused the error

- The XPDL file of the process definition that caused the error
- Information about Java Actions, Timers and Agents defined in the process definition
- Screenshot of the process instance history if the process instance goes into error state
- Stack trace if any exception is displayed

You can obtain the stack trace by clicking **Details** on the error page displayed in the Interstage BPM Console.

- Screenshot of the exception wherever it is displayed
- Screenshot of the process instance (graphical view) if the process instance goes into error state or into an unexpected state
- Calendar files (\*.cal) if timers are used
- The `agentsConfig.xml` file if agents are used

---

## Glossary

<b>ACID properties</b>	A transaction is a set of actions that obeys the four so-called ACID properties: atomic, consistent, isolated, and durable.
<b>Activity</b>	The description of a task, logical step, or work to be performed in a process. An activity is represented by a work item.
<b>Activity Time</b>	The time it takes to perform a particular activity.
<b>Agents</b>	Components that asynchronously access systems external to Interstage BPM.
<b>Annotation</b>	An addition to a process definition allowing for adding explanatory comments to the process definition.
<b>API</b>	Application Programming Interface. The interfaces and classes that programmers may use in their own customized applications to access the server.
<b>Application Variable</b>	Dynamic variables that can be defined at application project level to share them across all the processes within a specific application project. This saves effort for creating separate variables for each process.
<b>ASAP</b>	Asynchronous Service Access Protocol. ASAP is a communication protocol based on SOAP and is used to start, manage, and monitor long running services.
<b>Arrow</b>	A connector between nodes. Arrows guide the process flow from one node to another.
<b>Assignee</b>	The person(s) assigned to perform an activity.
<b>Attachment</b>	A document file generated by any application, which has been associated with a process.
<b>BPR</b>	Business Process Reengineering. The field of study which concentrates on how work may be redefined in terms of processes.
<b>Business Calendar</b>	A calendar that specifies working days and times.
<b>Business Process</b>	See <i>Process</i> .
<b>Business Rule Task Node</b>	A node that represents calling Business Rule Engine.
<b>Call Activity Node</b>	A node that represents a complex task. The details of that task are defined in another process definition.
<b>Chained-Process Node</b>	A node representing a complex task that can be accomplished independently from the tasks defined in the parent process definition.
<b>Compensation Action</b>	A second level Java Action that can be defined as compensation for a first level Java Action, e.g. for cleaning up the system and ensuring a consistent state of external systems not participating in a transaction.
<b>Database Action</b>	A Java Action allowing for the interaction with external database.
<b>DB Node</b>	A node that accesses an external database using JDBC.

---

<b>Directory Service (DS)</b>	Repository for the entire network's authentication and configuration data. Provides access to services, file servers, databases, and other applications. User and application access to the repository is controlled.
<b>Document Management System (DMS)</b>	The system integrated with Interstage BPM which is used to store attachments, forms, etc. The DMS Adapter is the communication link between the DMS and Interstage BPM.
<b>Due Date</b>	Specifies when an activity is due to be completed once it has become active. A due date also specifies what will happen when it is reached and the activity has not been completed.
<b>EJB</b>	Enterprise JavaBeans.
<b>Email Node</b>	A node that sends out predefined emails.
<b>End Node</b>	A node that identifies the end of a process branch and completes the process. A process definition has at least one End Node.
<b>Error Action</b>	A Java Action that is used to handle specific errors on process definition level, on remote subprocess level, and on first level Java Action.
<b>Flexible Exclusive Gateway Node</b>	A Simple Exclusive Gateway Node where the condition is specified as a JavaScript expression.
<b>Form</b>	An HTML or XML file which may be associated with an activity, process instance, or process definition. Forms can be created using Interstage BPM; their appearance can be modified using any XML or HTML editing tool.
<b>Framework Adapter</b>	The Framework Adapter, also called DD Adapter, connects the Directory Service and the Document Management System Adapters. The "DD" is short for "document" and "directory". It handles authentication of the user and manages a consistent user authentication to the Directory Services and Document Management System.
<b>Future Work Item</b>	Work Items that the users may be assigned in future so that the tasks can be planned in advance.
<b>Groups</b>	Collections of users. Groups can be defined in Interstage BPM's local group store, in a Directory Service or in both systems.
<b>Groupware</b>	A type of software, which facilitates collaboration.
<b>GUI</b>	Graphical User Interface.
<b>Initiator</b>	The person who starts a new process instance.
<b>Integration Action</b>	A Java Action allowing for accessing external functions from within a process definition.
<b>Interstage BPM Console</b>	A web-based user interface which allows a user to create process instances, process definitions and access and respond to work items. It is also used by Interstage BPM Super Users to administrate Interstage BPM.
<b>Interstage BPM Form</b>	A type of form native to Interstage BPM
<b>Iterator Node</b>	A node that generates multiple node instances upon specifying the iterator count.

---

<b>Java Action</b>	A part of a process definition that tells how to call a Java method. Every Java Action is an instance of a particular Java Action Type.
<b>Java Action Set</b>	A collection of Java Actions which are all executed at a particular time within a process definition
<b>Java Action Type</b>	This specifies what exact operation that a particular Java Action will call. A Java Action Type corresponds to a particular static Java method. You can extend the server by adding new custom Java Action Types, and that means that you are adding your custom Java methods to be called by the server, usually as part of an application.
<b>LDAP</b>	Lightweight Directory Access Protocol.
<b>Message Receive Node</b>	A node that is designed to wait for accepting an external message. It's the same action as a Receive Task Node.
<b>Node</b>	A graphical representation of a step in a process. Interstage BPM supports different node types, e.g. User Task Nodes, Parallel Join Gateway Notes, Call Activity Nodes, Simple Exclusive Gateway Nodes.
<b>No-Operation Action</b>	A built-in Java Action that specifies no operation.
<b>Notification Action</b>	A Java Action allowing for notifying users on events related to process execution. Users can be notified by email, for example, that a process or a single activity has been started.
<b>OnAbort Action Set</b>	Java Actions in this set will be executed before a process instance is aborted.
<b>OnResume Action Set</b>	Java Actions in this set will be executed before a process instance is resumed.
<b>OnSuspend Action Set</b>	Java Actions in this set will be executed before a process instance is suspended.
<b>Owner</b>	See <i>Process Definition Owner</i> and <i>Process Instance Owner</i> .
<b>Parallel Join Gateway Node</b>	A node that synchronizes flow from multiple branches in a process.
<b>Parallel Split Gateway Node</b>	A node that splits process flow into multiple parallel branches.
<b>Participant</b>	A person involved in a process.
<b>Process</b>	A sequence of steps that are performed to reach a business goal. Processes are modeled in process definitions.
<b>Process Definition</b>	The representation of a business process in a form that supports automated manipulation. A process definition defines the behavior and properties of the process instances created from it including the flow of control within the process.
<b>Process Definition Owner</b>	The person who created (or last edited) the process definition.
<b>Process Initiator</b>	See <i>Initiator</i> .

---

---

<b>Process Instance</b>	Represents a single enactment of a specific process definition. The structure of a process instance is exactly the same as the structure of the process definition on which it is based.
<b>Process Instance Due Date</b>	Represents the due date for a process instance.
<b>Process Instance Owner</b>	By default, the owner of a process instance is the owner of the process definition from which the process instance was created.
<b>Process Participant</b>	See <i>Participant</i> .
<b>Project</b>	A container for process definitions, forms, simulation scenarios, attachments, etc. On file system level, a project corresponds to a folder.
<b>QuickForm</b>	A structured, field-based HTML file created using the Interstage BPM Studio.
<b>Receive Task Node</b>	A node that is designed to wait for accepting an external message. It's the same action as a Message Receive Node.
<b>Remote Sub-Process Node</b>	A node representing a subprocess that runs on a remote workflow server.
<b>Role</b>	A relationship between a user or group and an object or context. This term is problematic since it has been used inconsistently in the past in the marketplace. A User Task node has an "Assignee". A group called "Window Washers" may be designated as the "Assignee" of the node. A user "Fred" may be in the group "Window Washers". This allows Fred to play the Assignee role (to be the assignee) when the node is active. The "Assignee Role" shows the relationship between Fred and the User Task node. Fred may also be a Process Owner for the process. "Process Owner" is another role, and users can play more than one role at a time. Remember that a role is a relationship between a person and a thing. For example, one person may drive a car, and "driver" is a role that person is playing with respect to the car. The things you find in a directory server are "groups". For example, a directory might have a group called "Drivers" and that is a group of people who might be allowed to drive, or who have the skill to drive, but it usually does not tell you who is currently driving. It is very rare for a directory to actually specify what person is currently playing a role.
<b>Rule</b>	Method used to determine choices on activities for which the rules are defined.
<b>SaaS Mode</b>	The 'Software as a Service' mode of Interstage BPM; deploying and using Interstage BPM in this mode allows you to create multiple tenants, and lease out Interstage BPM to these tenant organizations, who will use it as a service.
<b>Script Task Node</b>	A node that represents calling a script.
<b>Server</b>	In the Interstage BPM context, the component of the workflow management system installed on the computer to provides the run-time environment for a process.
<b>Server Action</b>	A Java Action allowing for interacting with the Interstage BPM Server.

---

---

<b>Service Task Node</b>	A node that represents using a service (an automated application).
<b>Simple Exclusive Gateway Node</b>	A node that directs the process flow along one of several branches, depending on specified criteria. Also known as an XOR Gateway.
<b>Simulation Scenario</b>	A defined setup for simulating the execution of a process definition on your local computer.
<b>SOAP</b>	Simple Object Access Protocol. SOAP is a standard communication protocol that allows one application to send an XML message to another application. It is used, for example, to access Web Services.
<b>SQL</b>	Structured Query Language.
<b>Start Node</b>	A node that identifies the beginning of a process. Every process definition has one and only one Start Node.
<b>SWAP</b>	Simple Workflow Access Protocol. SWAP passes XML messages over HTTP between workflow servers.
<b>Swimlane</b>	Visual grouping of activities.
<b>Task</b>	Same as an activity; a step in the process that requires a human response and usually a decision to be made.
<b>Timer</b>	Expires after a specified interval or at a specified time and date. Timers trigger certain actions when they expire.
<b>Timer Node</b>	A node that suspends process execution for a certain amount of time.
<b>User Defined Attribute (UDA)</b>	Data that process participants need to access, modify, or add, such as customer data, order numbers etc. User Defined Attributes are specified in the process definition at design time, and their values can be manipulated at run time.
<b>User Profile</b>	User-specific configuration information. This includes information such as whether a users wishes to receive email notifications, email address, and default directory, etc.
<b>User Task Node</b>	A graphical representation of a node.
<b>Voting Rule</b>	Rule defined on a Voting User Task Node to determine the outcome of the vote.
<b>Voting User Task Node</b>	A node that allows users to work on an activity in collaboration with one another.
<b>Web Service Node</b>	A node that retrieves data from a Web Service and makes it available for further processing.
<b>Workflow</b>	The sequence of activities within a business process.
<b>Workflow Application</b>	A bundle of artifacts such as process definitions, forms, simulation scenarios, attachments, etc. that are packaged together to form a process solution. Interstage BPM allows for the creation of Workflow Application projects having a predefined structure. Such applications can be setup on an Interstage BPM server in a single action.
<b>Workflow Server</b>	Same as a BPM Server: A server that provides the run-time environment for process instances.

---



<b>Work Item</b>	An assignment of a particular task to a particular user. Appears in a worklist.
<b>Worklist</b>	A list of workitems saying what tasks are currently assigned to which users.
<b>WSDL</b>	Web Services Description Language. WSDL is an XML-based language that describes the Web Services an organization offers. It also describes how to access the Web Services.
<b>XML Action</b>	A Java Action that performs specific operations on UDAs of type XML, for example adding XML substructures, setting text or attribute values in XML, or extracting UDA values from XML data.
<b>XPath</b>	XML Path Language. XPath is a language for finding information in an XML document. It is used to navigate through elements and attributes.
<b>XPDL</b>	XML Process Definition Language

---

# Index

**A**

About this manual, 9  
Architecture, 12, 13  
ASAP, 32  
Asynchronous Service Access Protocol, 32

**B**

Backup, 75  
Business Calendar, 35

**C**

Configuration export script, 25  
Configuration import script, 25  
Connectivity, 16  
Custom EJB, 15

**D**

Database  
    changing, 28  
Database connectivity, 16  
Directory Service  
    changing, 28  
Directory Service connectivity, 16  
DMS Root directories, 30  
DmsCollections.xml, 75

**E**

Enactment Engine, 14  
External systems, 17

**G**

Glossary, 124  
Group management, 15

**I**

Interstage BPM  
    Component architecture, 12, 13  
    documentation list, 10  
    Key features, 12

Interstage BPM (*continued*)  
    Overview, 12

**J**

Java Actions, 17  
JDBC, 16

**K**

kpilistener.conf, 75

**L**

LDAP, 16  
Log Files, 39

**M**

Model API, 17

**P**

Process Definition EJB, 14  
Process Instance EJB, 14

**S**

Simple Workflow Access Protocol, 32  
SWAP, 32

**T**

Tenant Management Console, 24  
Troubleshooting  
    Interstage BPM Server startup, 118

**U**

UDDI  
    Integration, 17  
User Agent EJB, 14  
User management, 15

**W**

Web Services, 17