Informatica

Informatica® ActiveVOS
9.2.4.6

Administration Console

Informatica ActiveVOS Administration Console
9.2.4.6
March 2020

Publication Date: 2023-11-23

Table of Contents

Preface

This module contains information on how monitor deployed assets and how to administer Informatica ActiveVOS. This is the start of the concept.

CHAPTER 1

Process Console Introduction

Refer to the topics below for some information to get started using Process Console.

How To’s

Refer to the following topics to learn about options commonly used in Process Console:

Process Console Basics
Starting and stopping the Process Server
Using Search
Importing and exporting a server configuration file
Deployment How To’s
Deploy processes
View and manage contributions
View deployed processes
View deployed services
Service Setup How To’s
Set up Identity service
Set up an email service
Set up JMS messaging service
Add an alert service for engine alerts
Add a monitoring service
Configure Process Central settings
Active Processes How To’s

View active processes
View a running, suspended or faulted process
Analyze problems from an execution log
Suspend, fix, and resume a process
Fix a running process migrated to a new process version
Use viewing filters
View and monitor tasks from People activities
Configure how many simultaneous requests can run concurrently for process throttling control
Process Server Configuration How To's
Set server properties
Set logging properties
Map URNS to URLS
Add global custom function contexts
Process Maintenance How To's
Work with process versions
Clean out old processes
Scheduling process execution
Process Server Maintenance How To's
Scheduling database and logs deletion
Server monitoring
Server monitoring Metrics

Deploy

The Deploy dialog allows you to add a business process archive (.bpr file) to the Cloud Server or a Secure Agent using Application Integration Console. The .bpr file, created in Process Developer, is similar to a web archive and contains all the resources needed to deploy your project resources so they are available at runtime.

You deploy one .bpr archive at a time, but the .bpr can include all the BPEL files, deployment descriptors, partner definition files, WSDL definitions, and other resources that you need.

To Deploy the .bpr file:

1. Log in to Informatica Intelligent Cloud Services using an account that has administrative privileges.
2. Select Application Integration Console
3. Select Deployed Assets.
4. In the upper-right corner of the page, click **Deploy**.
5. Click **Browse** to select your `.bpr` file.
6. Choose the Cloud Server or your Secure Agent from the **Deploy Target** list.
7. Click **Deploy**.

The server validates the files contained in the .bpr and stores the files in the database. The Logs page shows any errors, warnings, and information about the deployed process files.

After you deploy a .bpr file, you can view details for deployment descriptors, partner definition files, BPEL files, indexed properties, WSDL definitions, and schema files by accessing the specific resource in the Deployed Assets page. Catalog. .

Search

The Search feature allows you to quickly find an active process instance or catalog resource. Search includes an auto-complete feature.

If you enter a Process ID, you see the process instance. Select a process to see more details about. The details displays in a new browser tab or window.

Options for Different Process Servers

Process Server works with a variety of application servers, and some of the settings in the Process Console do not affect all application servers.

Note that special features for your application server are called out separately in online help.

Understanding the Properties of an Invoked Service

Process Server can invoke the services within a BPEL process using a variety of addressing options. In most cases, these options are defined for each partner role in the Process Deployment Descriptor (PDD) file that is deployed with a BPEL process. Alternately, addressing may occur dynamically within the process. In addition, some addressing options, such as service retry and security policies may be specified in a deployed BPEL file.

You can view the address properties of an invoked service on the Deployed Process *Version Detail* page. Under the Linkage column for a partner role, select the Static link to see the details.

For a detailed discussion of addressing options, refer to the *Process Developer Help*.

CHAPTER 2

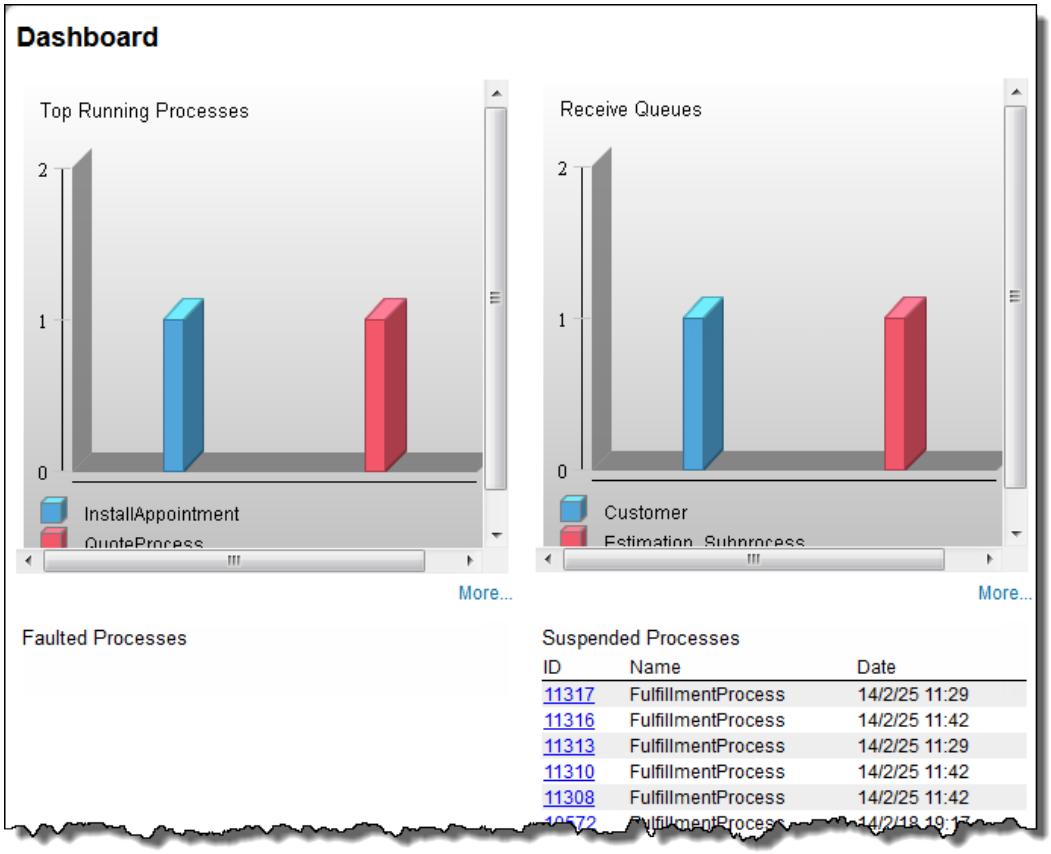
Home

The Home page of the Process Console links you to the following:

- Dashboard
- Active Processes. see
- Server Status

Dashboard

The information displayed on the Dashboard is real-time data from running and completed processes. The Dashboard is a snapshot of active processes displayed in a graphical report format. You can see graphs and charts of the most active processes as well as faulted and suspended processes and other data.

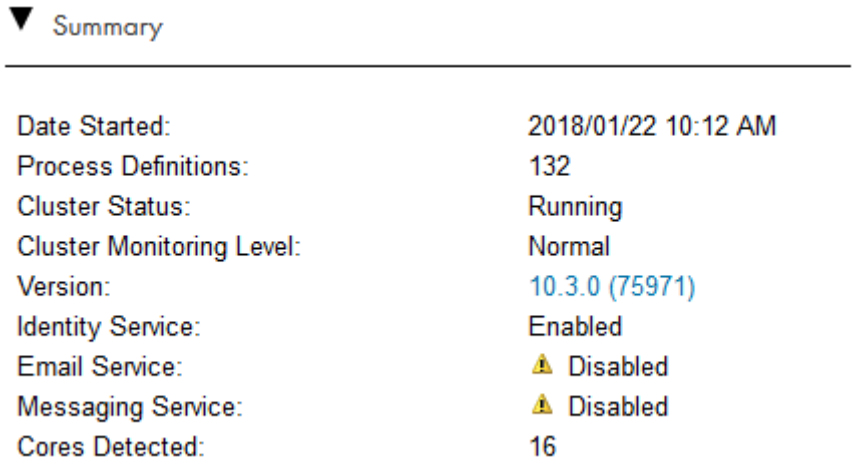


To use the Dashboard:

- Select **Refresh Reports** to update the data. You need to refresh only once every 30 minutes. The reports are cached for 30 minutes.
- Select **More** at the bottom of each graph to link to the Reports page for details

Server Status

The Summary tab provides an overview of the engine that executes BPEL processes.

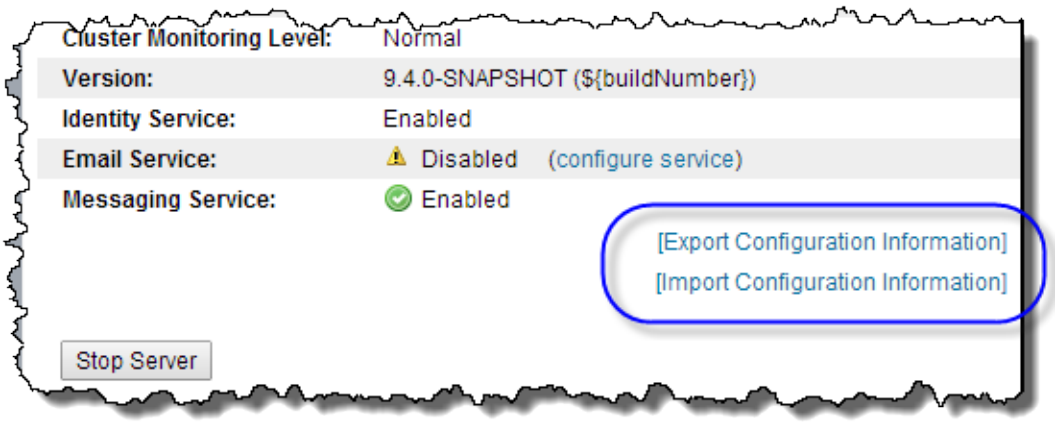


It contains the following items.

Item	Description
Date Started	Engine start date
Process Definitions	Number of business processes (`.bpel` files) currently stored in the database
Status (or Cluster Status)	Possible statuses values for Process Server engines are Running and Stopped. Additional database messages may also display. See *Admin* > *Maintenance* > *Storage* to see more detailed information regarding the database. The status displayed reflects all engines in a cluster. For example, if the cluster has two engines, and one is stopped, the status displays `Running (1/2 running)`.
Monitoring Level (or Cluster Monitoring Level)	Level indicates a monitoring severity; that is, it indicates whether a warning or error is detected. If there is no monitoring set up or if the engine is running normally, the level is Normal. Levels include Normal, Warning, and Error.
Version	Process Server version number
Identity Service Email Service Messaging Service	A status that indicates whether a service was configured and enabled.

Exporting and Importing Configuration Information

You can export and import an XML file containing information about Process Server properties and settings. You may also want to import a server configuration file created on one server into another serve so you can make both have the same properties and settings.



The properties and settings in this file include:

- URN Mappings. See "URN Mappings " on page 66.
- Schedules, including Database Purge Schedule and Scheduled Processes
- Configured Services, such as Alert Service and Messaging Service
- Process Server Properties and system information, such as engine status, Process Server log level, and process idle timeout

Note: You may be asked to export the file for technical support.

Stopping and Starting the Cluster

Starting and stopping the Process Server starts and stops all engines in the cluster. To start and stop individual engines in the cluster, see **Admin > Cluster**.

Process Server

The engine starts when you start your Process Server. The engine stops when you shut down the Process Server. Select **Stop Cluster** to stop the engine from accepting new requests and to save currently running processes. Select **Start Cluster** to change status to Running.

Process Server for Apache Tomcat

The Process Server engine starts when you start Tomcat; that is, it starts automatically. Process Server stops when you shut down Tomcat. Select **Stop Cluster** to stop the engine from accepting new requests and to save currently running processes. Select **Start Cluster** to change the engine status to Running

License Warning

The License Warning field displays messages to help you stay in compliance with your Informatica Business Process Manager license agreements. The Process Server can verify that the number of installed licenses matches the configuration of your server cluster.

To check for license/CPU compliance in Process Server, see **Admin > Cluster**.

To Check License/CPU Compliance in Process Server for Apache Tomcat

1. Select the **Admin > License > CPU Count**.
2. Type in the number of CPUs on the server where Process Server is installed, and select **Update**.
3. Select the **Home** menu.

If the number of CPUs on the server exceeds the number of licenses installed, a warning is displayed on this page.

CHAPTER 3

Monitor

This chapter includes the following topics:

Process Monitoring

The Monitor menu in the Process Console provides access to process, task, and server monitoring options that include:

- **Active Process**: View the list of process instances that have been or are currently executing.
- **Alarm Queue**: View a list of active 'On Alarm' Process activities.
- **Receive Queue**: View a list of active receive, onMessage, and onEvent activities queued for incoming messages.
- **Dispatch Service**: View details for services governed by the `DispatchThrottle` policy. You see this option only if you use an agent console.

Active Processes

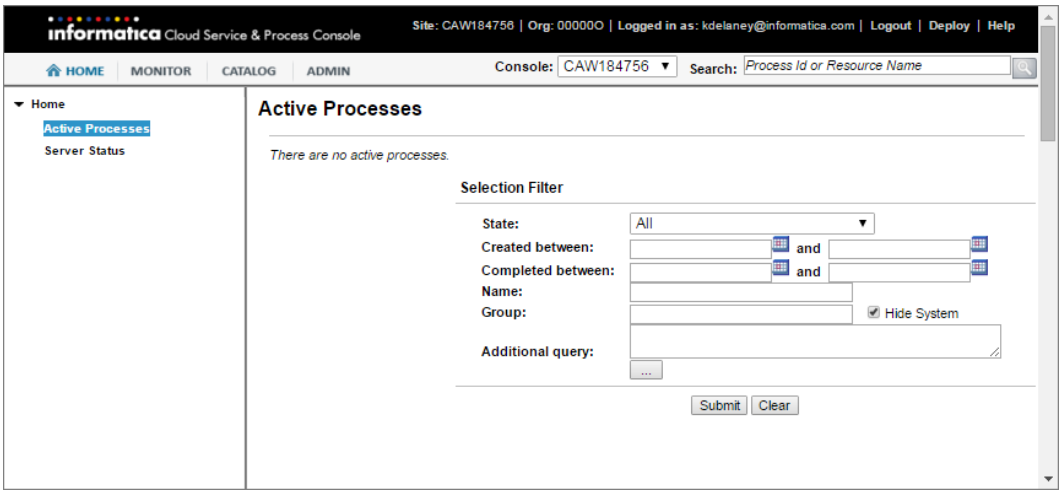
The Active Processes page shows a list of process instances that are executing or completed, along with the process version. States can be Running, Suspended, Completed, Compensatable (for a subprocess), or Faulted.

From the Monitor menu, choose **Process Monitoring** > **Active Processes.**

Select filter criteria to select specific processes. For a Secure Agent, you can also show/hide system processes. For the Cloud Server, you have an option to hide/show both system and public processes.

Click **Submit** to view all processes or those that meet the filter criteria, if any.

After you submit the selection options, a list of Active Processes displays, as shown here for a Secure Agent:



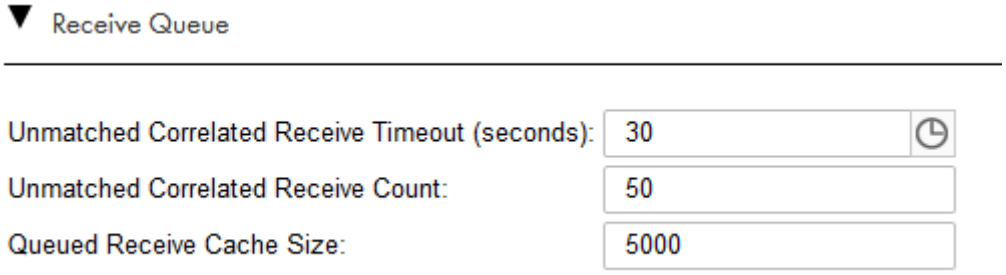
Alarm Queue

Under **Selection Filter**, select these Alarm Queue options to determine which process activities appear when you view active alarms in the On Alarm list:

Option	Description
Deadline Between	Beginning and Ending date and time for alarm.
Process ID	Process instance ID. You can find this ID on the Processes page.
Process Name	Local part of the process-qualified name (`qname`) .
Group	The Group this process belongs to (optional). The Group name displays in the On Alarm list.
Hide System	Uncheck to display the default System processes. The System processes are hidden by default.

Receive Queue

Use the controls on this page to set the size for how many queued receives to hold in memory to reduce the number of queries to the database. Sizing depends on server memory constraints.



The information on this page is as follows:

Unmatched Correlated Receive Timeout

The amount of time (in seconds) that the engine waits for a correlated message to be matched to a receive activity. This is only if the message arrives before the activity becomes active. If a correlated message takes longer than the specified time to be matched, the engine discards the message and you see a correlation violation exception

The default value is 30 seconds. To avoid many unconsumed messages on the server, set this value to a low number. If you set this value to specify zero seconds, the engine discards all unmatched correlated messages immediately.

When you click the **Clock icon**, you see the duration-chooser dialog box.

For more details, see *Server Properties*.

Unmatched Correlated Receive Count

The maximum number of unmatched correlated receives the engine will hold in memory at a time. When this limit is crossed, the engine rejects unmatched correlated receives; starting from the oldest, to make room for newer receives.

Queued Receive Cache Size

The size of the cache memory that holds activity information when the system is busy with other requests.

You can also change any of the values displayed for these three fields.

If events are displayed, select a receive and then select a partner link to view details. A window opens where you can see the BPEL process location in which the receive activity executes. You can also see the correlation property alias and data, if any, associated with this receive activity

Dispatch Queue

Use the dispatch service to create dispatch configurations that throttle requests made to Process Server.

The requests could be HTTP/S messages, message events, or other Transport Layer Security (TLS) requests. You can queue requests, dispatch requests in batches, and control the maximum number of requests that run concurrently.

For example, you have a frequently used process that quickly dispatches asynchronous requests to Process Server. At run time, if thousands of these requests execute simultaneously, the result can overwhelm the server and the service being invoked. Use the dispatch service to send requests in batches and to ensure that only a few requests are processed at the same time.

You can use the dispatch service for services that run on the Secure Agent. You cannot use the dispatch service for services that run on the cloud server.

Create a dispatch configuration for a service. At run time, Application Integration chooses the dispatch configuration in the following order of preference:

1. Service Name
2. System Default

For example, the service `processA_event1` has the following dispatch configurations that apply to it:

- `processA_event1` (Service Name)
- `SystemDefault`

When Process Server executes `processA_event1`, the `processA_event1` dispatch configuration applies.

The dispatch service applies to all requests except for subprocess invokes that contain boundary events. For example, if a subprocess step contains a message event, the dispatch service will not apply to the subprocess.

However, if you use a service call step to add a subprocess, the dispatch applies to all subprocess configurations, including subprocesses with boundary events.

Default and Custom Dispatch Services

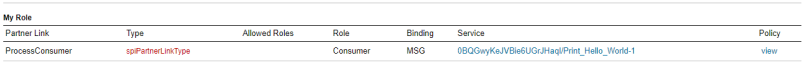
Process Server has default dispatch services.

The following table shows the default dispatch services and their purpose:

Default Dispatch Service	Purpose
SystemDefault	Applies to all services.
avBusinessConnectionRESTService	Applies to service connectors.
avCreateAnyEntityService	Enables the host runtime environment to PUT records using a database connection.
avDeleteAnyEntityService	Enables the host runtime environment to DELETE records using a database connection.
avHostEnvironmentRuntimeAccess	Enables the host runtime environment to access records using a database connection.
avHostRuntimeCall	Enables the host runtime environment to GET records using a database connection.
avProxyToProcess	Enables a process to call other processes as a services.
jmsEnqueueService	Applies to the JMS automated step.
sfHostEnvironmentRuntimeService	Enables the host runtime environment to access a Salesforce database using a Salesforce connection.

Dispatch Service Configuration

Configure the following properties to create a custom dispatch configuration:

Property	Description
Name	The name of a service definition. To view a service definition name, go to **Processes** > **Select Process Version Number** > **My Role** > **Service column**. For example, in the following **My Role** section of the **Deployed Process Version Details** page, the service name is `0BQGwyKeJVBie6UGrJHaqI/Print_Hello_World-1`:  My Role Partner Link: ProcessConsumer Type: spiPartnerLinkType Allowed Roles: Role: Consumer Binding: MSG Service: 0BQGwyKeJVBie6UGrJHaqI/Print_Hello_World-1 Policy: view If you enter a service definition name, the dispatch configuration governs all instances of the service, regardless of the process that contains the service. You cannot change the name of a saved dispatch configuration. To edit the name, delete the dispatch configuration and then create a new one.
Max Concurrent	The maximum number of requests that Process Server can hold in the concurrent pool. Default is 15.
Max In-Memory	The maximum number of requests that Process Server can hold in the in-memory queue. If you enable persistence, the dispatch service rejects requests that exceed the **Max Queued** value. If you do not enable persistence, the dispatch service rejects requests that exceed the **Max In-Memory** value. Default is 300.
Max Queued	The maximum number of requests in the persistent queue. If you enable persistence, the dispatch queue service rejects requests that exceed the **Max Queued** value. If you do not enable persistence, the dispatch service rejects requests that exceed the **Max In-Memory** value. Default is 300.
Timeout (seconds)	The maximum amount of time, in seconds, that a single request can take to execute within the dispatch service. If a request takes longer that the timeout value to execute, it moves out of the dispatch service and the next pending request executes. If a request times out of the dispatch service, Process Server continues to execute outside of the dispatch configuration. As a result, you might see different values on the **Processes** list and the **Server Configuration** > **Queues** > **Dispatch Queue** list. Default is 300 seconds.
Persistent	Select the **Persistent** option to save pending requests to the persistent queue. If you select this option and the Secure Agent restarts, the dispatch configuration continues to execute queued requests. If you do not select the **Persistent** option, pending requests are saved only in the in-memory queue and are lost if the Secure Agent restarts.

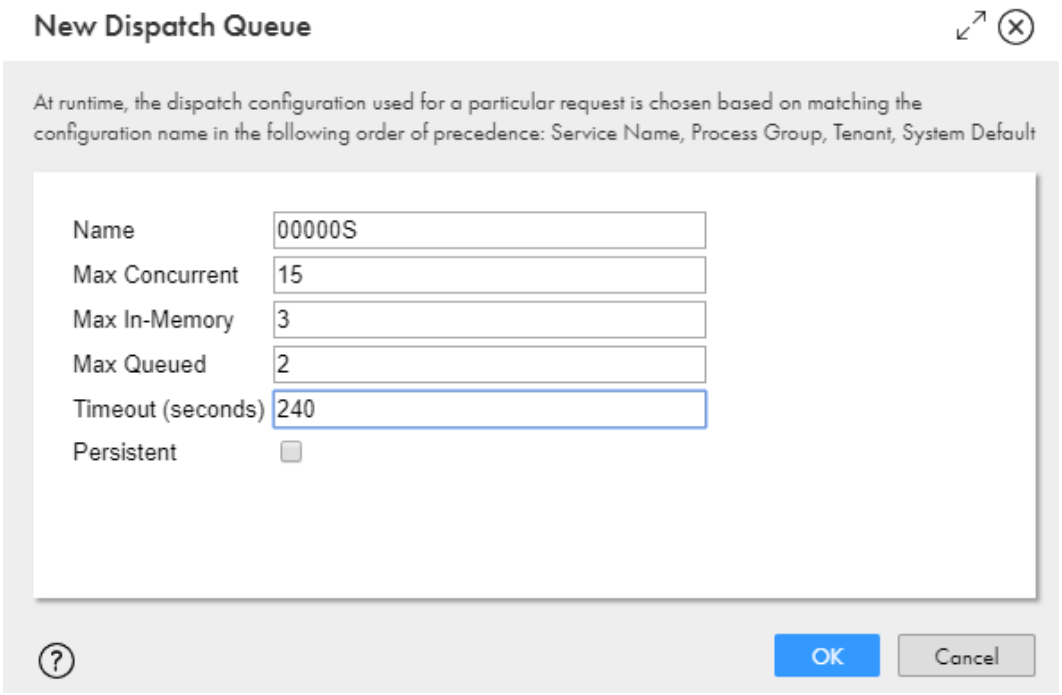
Creating a Custom Dispatch Configuration

Perform the following steps to create a custom dispatch configuration:

1. Log in to Application Integration Console.

2. Go to **Server Configuration** and select a Secure Agent.
3. Go to **Queues** > **Dispatch Queues** > **Add**.
4. In the **New Dispatch Queue** dialog box, configure properties and then click **OK**.

The following image shows a sample **New Dispatch Queue** dialog box for the 00000S service:



Task Monitoring

After selecting the Task Monitoring page, Process Server displays the following page.

Using the commands on this page, you can monitor:

Active Tasks

View the list of active tasks currently being processed by users.

Work Queues

View work queues by workgroups and tasks.

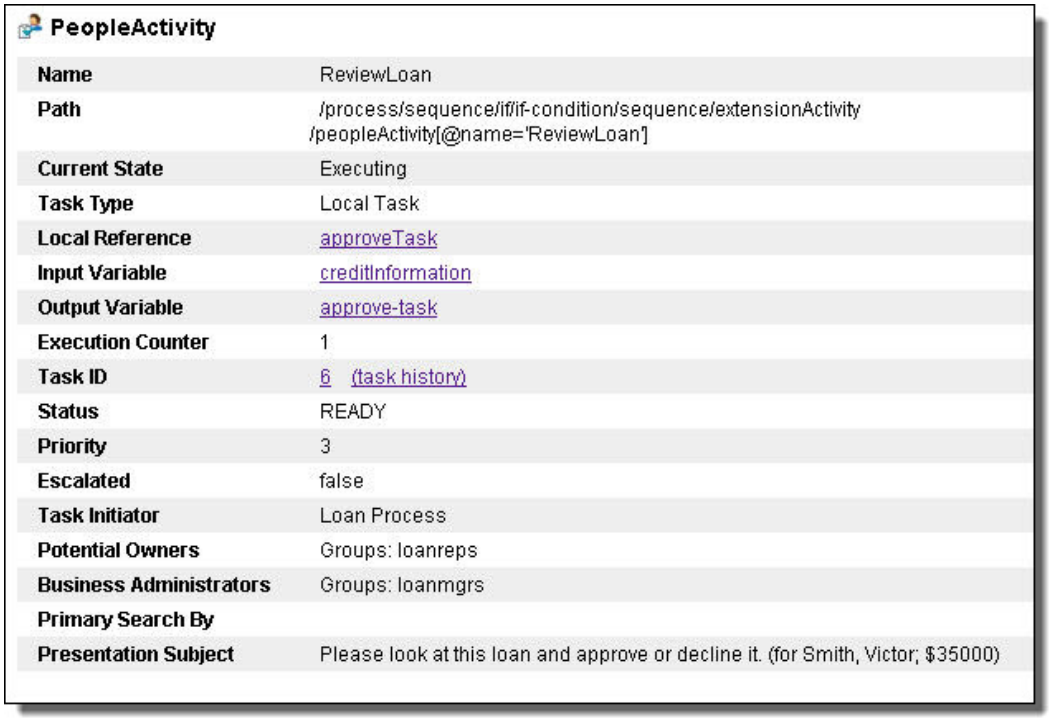
For an overview of this topic, open the *Process Developer Guide* and refer there to information on Human Tasks.

Human Tasks

Human Tasks is an extension to WS-BPEL 2.0 that includes human workflow activities within a BPEL process. In a BPEL process, a People activity "invokes" a person to handle and complete a task. Similar to the behavior of an invoke activity, the person returns output data to the process. The task input and output data is sent and received from Process Central (or via the WS-Human Task Web Service). Additionally, users can receive notifications, which contain only an input variable.

Viewing Human Task Information

From the Active Processes page, you can select a process to open the Active Process Detail page. In this page, select a People Activity from the graph or Outline view. The following illustration shows the properties for a People Activity that contains a task.



PeopleActivity

Name	ReviewLoan
Path	/process/sequence/if/if-condition/sequence/extensionActivity /peopleActivity[@name='ReviewLoan']
Current State	Executing
Task Type	Local Task
Local Reference	approveTask
Input Variable	creditInformation
Output Variable	approve-task
Execution Counter	1
Task ID	6 (task history)
Status	READY
Priority	3
Escalated	false
Task Initiator	Loan Process
Potential Owners	Groups: loanreps
Business Administrators	Groups: loanmgrs
Primary Search By	
Presentation Subject	Please look at this loan and approve or decline it. (for Smith, Victor; $35000)

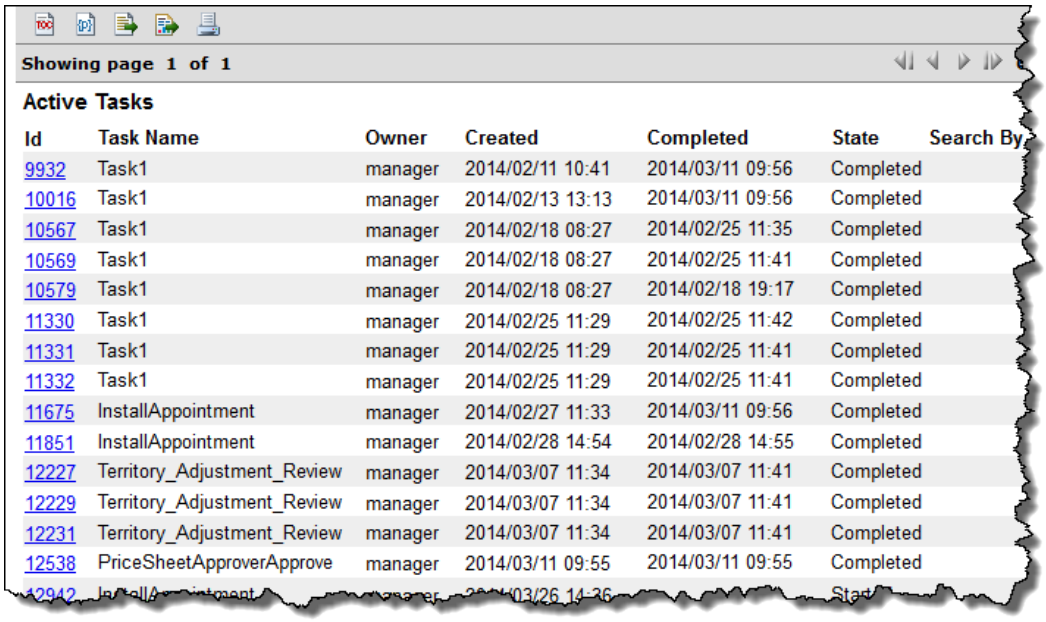
In this view you can select:

- Local Reference or Inline Reference to view the task or notification definition
- Input Variable or Output Variable to view the current data. Note that a notification has only an Input Variable.
- Task Id to open the Process Central Login page to sign in and take some action on the task
- Task history to view the current owner and work status of the task

In addition to looking at one task at a time, you can view a list of active tasks.

ActiveTasks

An active task is associated with a running or completed People activity. It contains generic human roles, such as potential owners and administrators which allow users to work on them. Typically this work is carried out using the Process Central, but the comprehensive WS-Human Task API allows the creation of custom applications as well.



Showing page 1 of 1

Active Tasks

Id	Task Name	Owner	Created	Completed	State	Search By
9932	Task1	manager	2014/02/11 10:41	2014/03/11 09:56	Completed	
10016	Task1	manager	2014/02/13 13:13	2014/03/11 09:56	Completed	
10567	Task1	manager	2014/02/18 08:27	2014/02/25 11:35	Completed	
10569	Task1	manager	2014/02/18 08:27	2014/02/25 11:41	Completed	
10579	Task1	manager	2014/02/18 08:27	2014/02/18 19:17	Completed	
11330	Task1	manager	2014/02/25 11:29	2014/02/25 11:42	Completed	
11331	Task1	manager	2014/02/25 11:29	2014/02/25 11:41	Completed	
11332	Task1	manager	2014/02/25 11:29	2014/02/25 11:41	Completed	
11675	InstallAppointment	manager	2014/02/27 11:33	2014/03/11 09:56	Completed	
11851	InstallAppointment	manager	2014/02/28 14:54	2014/02/28 14:55	Completed	
12227	Territory_Adjustment_Review	manager	2014/03/07 11:34	2014/03/07 11:41	Completed	
12229	Territory_Adjustment_Review	manager	2014/03/07 11:34	2014/03/07 11:41	Completed	
12231	Territory_Adjustment_Review	manager	2014/03/07 11:34	2014/03/07 11:41	Completed	
12538	PriceSheetApproverApprove	manager	2014/03/11 09:55	2014/03/11 09:55	Completed	

For a description of the Active Tasks Toolbar, see "Reports (Process Developer) " on page 47.

The following are the fields displayed for each active task

Field	Explanation
Id	Id assigned to the task by the engine. Select the Id to go to the Process Central login page. You can sign in as one of the users in the group to view and take action on the task.
Task Name	Name added to the task when it was created in Process Developer
Owner	User who claimed the task
Created	Time the People activity was executed
Completed	Time the user submitted output data or faulted the task or the task expired
State	Current lifecycle state such as started, claimed, reassigned
Search By	Search expression added to the task when the task was developed within Process Developer

Using the Selection Filter

Fill in one or more the following filter criteria:

- **State**. Select one of the task states shown in the table that follows this list.
- **Created between**. Date and time range for task starts.
- **Completed between**. Date and time range for task completions.

- **Name**. Task name.
- **Owner**. Current user who claimed a task and is working on it. Typically the owner has a role of Potential Owner.
- **Other Role**. Users who have been identified with a role within the task definition. Roles include Administrator, Initiator, Potential Owner, and Stakeholder.
- **Search By**. Expression in the task definition.

State	Meaning
All	All states of tasks
All Open	Claimed, unclaimed, and suspended tasks
All Closed	Completed, faulted, exited, and skipped tasks
All Problem	Tasks are faulted, exited, or error
Deferred	Tasks defined to begin at a certain time
Claimed	Tasks have owners and are in progress
Unclaimed	Tasks are ready, but do not have owners
Faulted	Tasks completed with a fault
Suspended	User-suspended tasks
Completed	Normal completions
Skipped	Tasks defined as "skippable" and were skipped by the task initiator
Exited	Tasks that reached a completion deadline without being completed
System Error	Task failed because of a system error

See also "Work Queues (Process Developer) " on page 23.

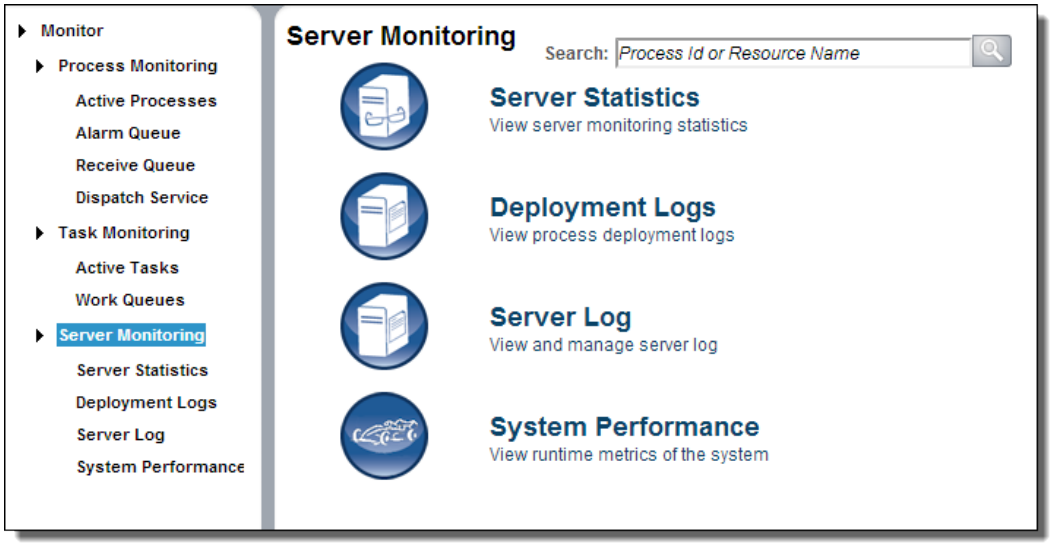
Work Queues (Process Developer)

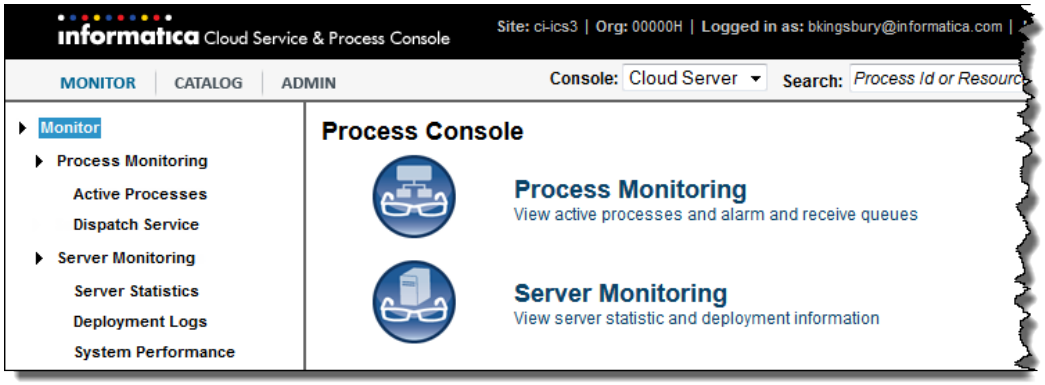
A work queue is a list of groups associated with unclaimed tasks. Under each group is a list of tasks. Each task has a count showing how many tasks are unclaimed.

You can select the unclaimed task count to go to the Active Tasks list. This list displays a filtered list of tasks of that type.

Server Monitoring

After pressing the Server Monitoring command, Process Server displays the following page:





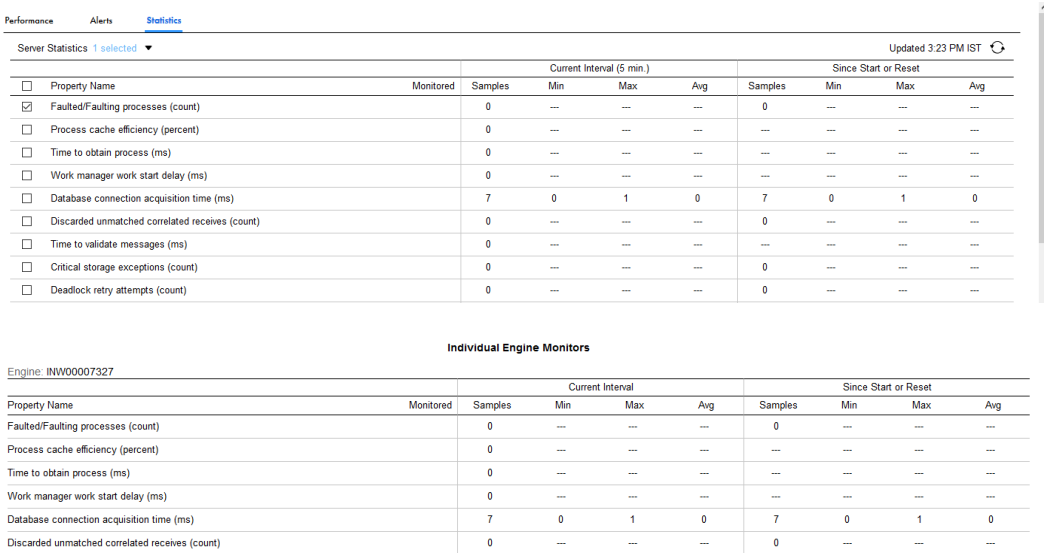
Process Server monitoring includes:

- Server Statistics: (Agent only) View server monitoring statistics
- Deployment Logs: View process deployment logs
- Server Log; View and manage the server log
- System Performance: (Agent only) View runtime metrics

Server Statistics

Note: You only see this command if you have selected an agent.

Use the Statistics page to view the monitoring properties you configured or which are the default. Process Server collects engine statistics and then aggregates them by intervals. If desired, you can configure a threshold interval and an error/warning level for each of these properties. If you configure these settings, you will see additional signals displayed on this page, such as an eyeglass icon indicating a property is being watched.



Performance Alerts Statistics

Server Statistics 1 selected

Updated 3:23 PM IST

Property Name	Monitored	Current Interval (5 min.) Samples	Min	Max	Avg	Since Start or Reset Samples	Min	Max	Avg
Faulted/Faulting processes (count)		0	---	---	---	0	---	---	---
Process cache efficiency (percent)		0	---	---	---	---	---	---	---
Time to obtain process (ms)		0	---	---	---	---	---	---	---
Work manager work start delay (ms)		0	---	---	---	---	---	---	---
Database connection acquisition time (ms)		7	0	1	0	7	0	1	0
Discarded unmatched correlated receives (count)		0	---	---	---	0	---	---	---
Time to validate messages (ms)		0	---	---	---	---	---	---	---
Critical storage exceptions (count)		0	---	---	---	0	---	---	---
Deadlock retry attempts (count)		0	---	---	---	0	---	---	---

Individual Engine Monitors

Engine: INW00007327

Property Name	Monitored	Current Interval Samples	Min	Max	Avg	Since Start or Reset Samples	Min	Max	Avg
Faulted/Faulting processes (count)		0	---	---	---	0	---	---	---
Process cache efficiency (percent)		0	---	---	---	---	---	---	---
Time to obtain process (ms)		0	---	---	---	---	---	---	---
Work manager work start delay (ms)		0	---	---	---	---	---	---	---
Database connection acquisition time (ms)		7	0	1	0	7	0	1	0
Discarded unmatched correlated receives (count)		0	---	---	---	0	---	---	---

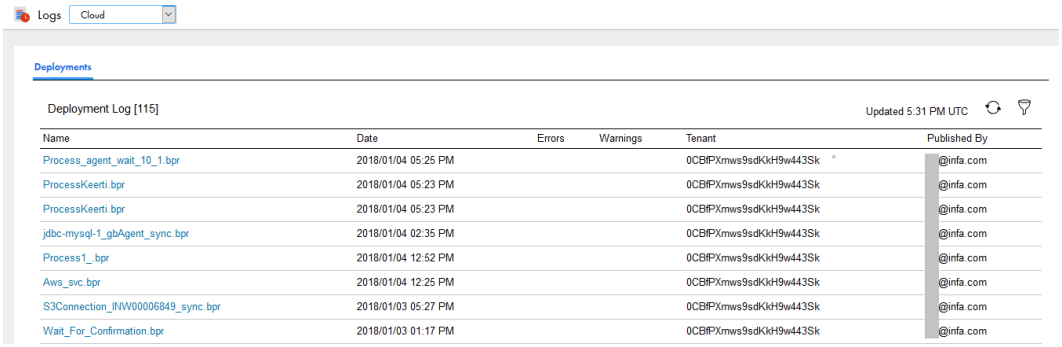
During a collection interval, Process Server maintains statistics for each configured property value, including the maximum, average, and total count values. In addition, historical statistics are collected. Historical statistics include minimum and maximum values for the intervals being recorded. Statistics are collected in-memory.

Select **Reset** to clear statistics. For example, if you correct an error condition, you can reset the statistics collection starting with zero values to see if performance improves. Once all errors and warnings are cleared, the engine monitoring level is reset to Normal.

Deployment Logs

The Deployment Logs tab shows a list of logs generated when new and modified business process archive (`.bpr`) files are deployed.

The following image shows a sample Deployment Logs tab :



Logs Cloud

Deployments

Deployment Log [115]

Updated 5:31 PM UTC

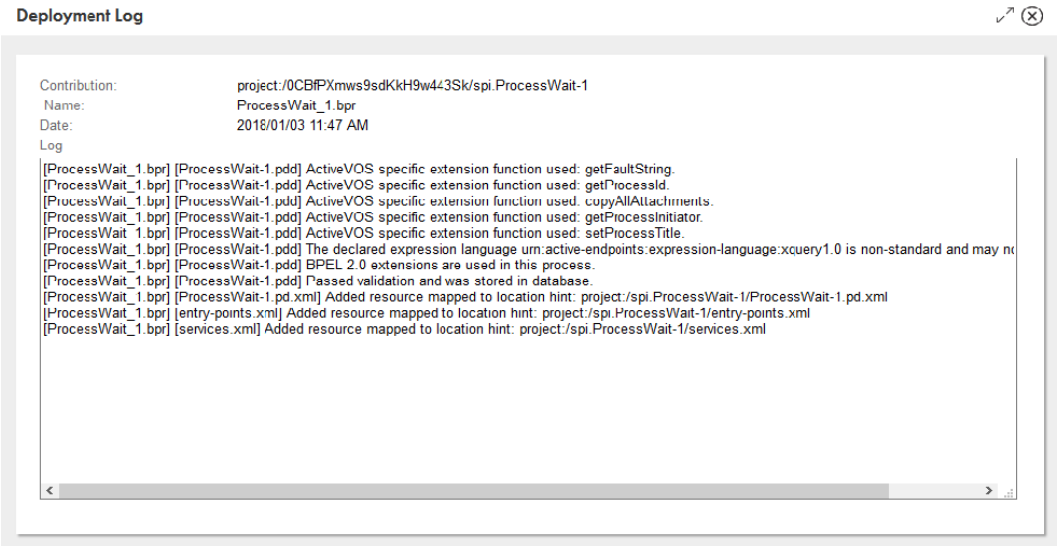
Name	Date	Errors	Warnings	Tenant	Published By
Process_agent_wait_10_1.bpr	2018/01/04 05:25 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com
ProcessKeerti.bpr	2018/01/04 05:23 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com
ProcessKeerti.bpr	2018/01/04 05:23 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com
jdbc-mysql-1_gbAgent_sync.bpr	2018/01/04 02:35 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com
Process1_.bpr	2018/01/04 12:52 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com
Aws_svc.bpr	2018/01/04 12:25 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com
S3Connection_INW00006849_sync.bpr	2018/01/03 05:27 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com
Wait_For_Confirmation.bpr	2018/01/03 01:17 PM			0CBfPXmws9sdKkH9w443Sk	@infa.com

The number of errors and warnings generated, if any, are shown.

On this page you can:

- Change the display of the logs list by using the Selection Filter
- Select a `.bpr` file to view its deployment log

After you click on a `.bpr` file, Process Developer displays its Deployment Log page. This page has the name, date, and log for the selected `.bpr` file.



During deployment, the engine validates the deployment descriptor of the BPEL process, ensuring that the its WSDL file and other resources are available and valid for the current version of the process. If validation errors or warnings occur, make corrections and redeploy the `.bpr` file or create a new `.bpr` file for invalid processes.

Note: When you delete a contribution, its deployment log is also deleted.

Selection Filter

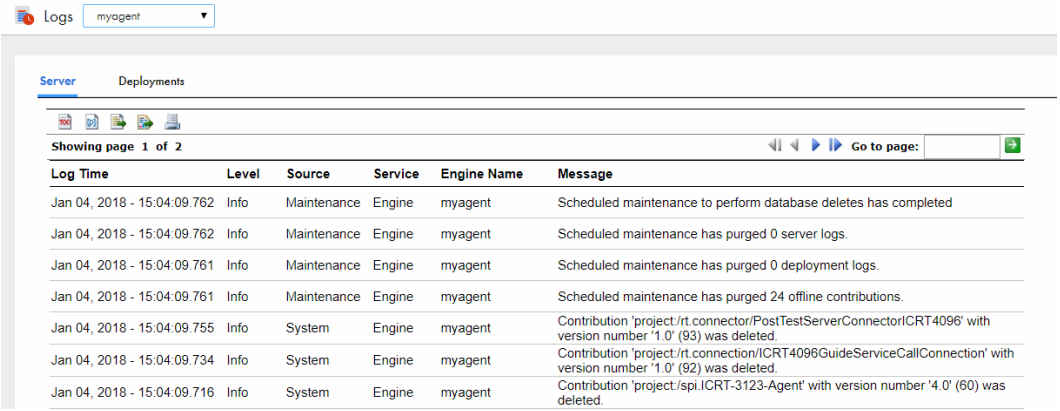
1. Select the Log Contents type, if desired.
2. Select Deployed between dates, if desired.
3. Type in the Name of a `.bpr` file, if desired. You can use a wildcard (*) in the name.
4. Select **Submit**. The Deployment Logs list redisplays based on your selection filters.

Server Log

The Server Log captures events occurring on the server so that you can view and troubleshoot problems. For each event, the log includes the time, problem level, source (initiator), service (event type), engine, and message describing the event. For example, a BPR deployment causes a log event with info level, system source, engine service.

Note: The server log may not update immediately after an event occurs. It is typically updated between 30 and 60 seconds after an event. If you are not seeing what you think should be there, you may just need to update your browser.

The following image shows a sample Server Log:



This topic discusses:

- Server Log Information
- Deleting the Process Server Log
- System Logging for WebLogic Users

Server Log Information

One of the following filters was used when generating logging information:

- **Log Time (Logged after and Logged before)**. By default, the log displays the most recent six hours of events. Refresh the page to view the most recent date and time. You can change the log period by setting the date and time. As a time shortcut, you can select the **Now** icon at the end of the time text box.
- **Level**. One of the following was selected:
  - Verbose—All levels
  - Info
  - Warning
  - Error
  - Critical
- **Source**. The initiator of the event or select All. Sources are:
  - System—Process Server events such as deployments
  - Maintenance—Storage and deployment log deletions
  - User—User-initiated
- **Service**. The services are:
  - Alert—Alert service configuration events
  - Email—Email service configuration events
  - Engine—Engine events, including deployments, Process Server start/stop, storage events, monitoring alerts
  - Identity—Identity service (file, LDAP, JDBC) configuration events
  - Messaging—Messaging service (JMS or other) configuration events
  - Task.—Client URI update events
  - Process—Messages added to Process Server Log from a process using the Process Server Log Comment system service

- All—All services are shown

- **Engine Name**. The engine upon which execution occurred.
- **Message** . Text describing the event.

The console records the User/IP address when a purge or scheduling request occurs. An application server may have its own method. Here, for example, is what might be shown for Tomcat: `https://tomcat.apache.org/tomcat-7.0-doc/api/org/apache/catalina/valves/RemoteIpValve.html`.

Deleting the Server Log

Use the **Clean Up Storage** on the **Storage** tab of the **Server Configuration** page.

In addition, the interface for clearing the server log is available in Process Developer as a system service so that a developer can create and deploy a process to delete the log programmatically.

System Logging for WebLogic Users

If you are using WebLogic, you must enable system logging from the WebLogic Process Console to have the Process Server `stdout` write to the WebLogic server log.

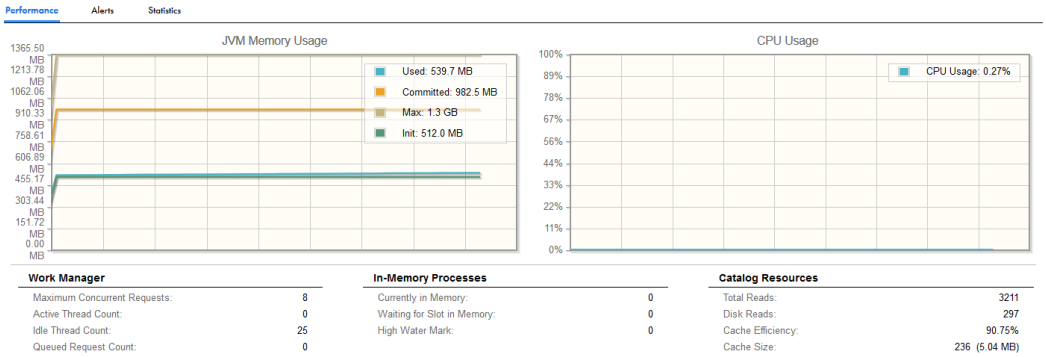
1. In the WebLogic Process Console, select the Logging top-level tab.
2. In the General sub-tab, enable Advanced options'
3. Select the **Lock & Edit** button so you can make changes.
4. Enable the Redirect stdout logging enabled checkbox.
5. Choose **Activate Changes** to apply your changes

System Performance

You see these details on the Performance tab:

- Overall performance data.
- Performance and throughput of service requests.
- Resource and service bottlenecks.

For example:



The statistics show a snapshot of what was in memory when you first open the page. Refresh the page to get updates.

These statistics let you ask questions such as:

- How many requests are being sent to the server?
- How long is the queue?

- How many requests are waiting?
- How long does it take to provide a response?
- How long is it taking for the server to return from service calls?
- Should I increase the work manager limits set on the Server Properties page?

After you run a load test, you can watch the metrics and set or reset configuration properties.

Platform Support

As you analyze performance in relation to your application server, keep the following in mind:

- Statistics important to consider before changing other settings include:
  - CPU Utilization—graph over time interval
  - JVM Memory Usage—graph over time interval
  - Database Connection Pool—available, in use, waiters, and high water marks

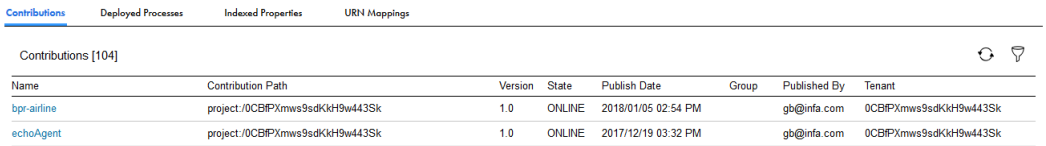
CHAPTER 4

Catalog, Reports, and Custom Faults

This chapter includes the following topics:

Contributions

The Contributions tab tab lists all business process archive (BPR) deployments. Their names are based upon the Process Developer project from which they were deployed.



Name	Contribution Path	Version	State	Publish Date	Group	Published By	Tenant
bpr-airline	project:/0CBfPXmws9sdKkH9w443Sk	1.0	ONLINE	2018/01/05 02:54 PM		gb@infa.com	0CBfPXmws9sdKkH9w443Sk
echoAgent	project:/0CBfPXmws9sdKkH9w443Sk	1.0	ONLINE	2017/12/19 03:32 PM		gb@infa.com	0CBfPXmws9sdKkH9w443Sk

Just as Process Developer keeps together all files in an orchestration project, a contribution is a unit of the deployable processes and resources from a project. It is a unique deployment to the server catalog. As developers update their projects, they can deploy new contributions while maintaining earlier ones on the server.

Here is some information about contributions:

- Each contribution is named (by default) as `project:/project name` as a Process Developer location hint.
- A contribution can contain multiple processes, which can be managed together or separately.

- Each new version of a contribution supersedes an earlier version and becomes the current (online) version. However, you can rollback an online contribution to an earlier version. This contribution tracking system allows for easy management of processes and their resources.
- There is only one current (online) contribution. All contributions can become the online version.
- The current (online) contribution can be rolled back to an earlier version.
- Process versions are associated with a specific version of WSDL, schema, and other resources. Every BPEL process is associated with the set of resources it is deployed with.
- If you delete a contribution, all associated files and the deployment log are deleted, giving you an easy way to clean out the database during testing
- Multiple developers can deploy the same process and resources without colliding with one another.

For details, see *Contribution Detail*.

What Happens When a Contribution is Updated

When a BPR contribution is deployed for the first time, Process Server tags the contribution and every file in the deployment as version 1.0, and the contribution is online. Subsequently, as developers make changes, they may update, add, or delete files and then repdeploy the same BPR. The new version becomes the online version and the old version is offline pending, eventually becoming offline when all executing processes complete.

The following scenarios illustrate what happens when a contribution is updated.

Contribution changes from V1 to V2 -- Process1 is modified, Process2 is removed

Deployment 1 project:/MyProject V1	**Deployment 2 project:/MyProject V2**	**Changes on the Server**
`process1.bpel`	`process1.bpel` (modified)	V2 is online, V1 is offline or offline pending, if it has associated process instances.
`process.private.wsdl` `process.public.wsdl`	not modified	V1.01 is online. V1 is offline. With each deployment, WSDL and XSD are incremented to avoid cache issues.
`process2.bpel`	(not included)	The process is no longer in the catalog.

Contribution changes from V1 to V2 -- All original files are removed, a new one is deployed

Deployment 1 project:/MyProject	**Deployment 2 project:/MyProject V2**	**Changes on the Server**
(`myAVOS.avcconfig` does not exist)	`myAVOS.avcconfig`	This file is deployed as an "Additional Resource" and is the only file deployed in the same BPR as previous deployment. V1 is online. This is the only file online because it is the only file in Contribution V2. To get all files back online, redeploy the BPR and include all files.
`process1.bpel`	(not deployed)	BPEL file is removed
`process.private.wsdl` `process.public.wsdl`	(not deployed)	WSDL files are removed

For each contribution, basic version information is displayed, as shown in the table.

Item	Description
Contribution	The base URI, usually the Process Developer project name, such as `project:/myProject`
Version	Deployed version
State	By default, online state is displayed; for information, see the description of all states below this table
Date	Date the contribution was deployed
Group	Group this contribution belongs to, if any, as added in the Export Business Process Wizard in Process Developer
Deployer	Name of person (or system) who deployed, if authentication is required for deployers. The default is anonymous.

Contribution Versions

Contribution versions can have one of the following states:

- Online—By default, the newest contribution (highest numbered) is the online (current) version.
- Offline Pending—A version is offline pending in the following cases: if you set the version offline and there are running process instances; a newer version is deployed while there are running process instances; or the contribution's exported namespaces are referenced by other contributions.
- Offline—When all process instances of an offline pending version complete, a contribution is offline.

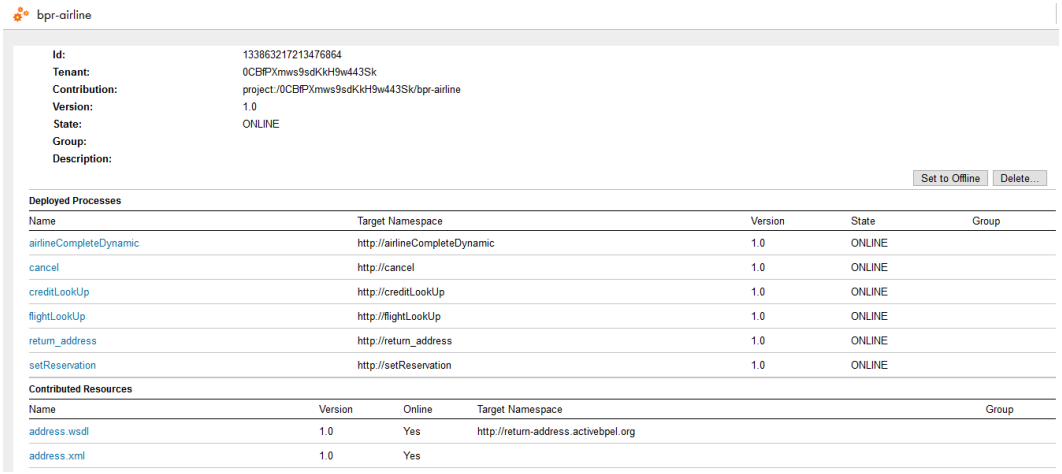
Selection Filters

You can select the following Selection Filters to view a subset of contributions:

- State—One of the contribution versions states just listed.
- Deployed between—Enter or select the starting and ending deployment dates.
- Name—Type in the contribution name. You can use wildcards, such as tutor*.*, and select **Submit**.
- Group—If the contribution was deployed with a group name, this filter is available. Type in a group name, and select **Submit**. There is a default System group, which is hidden by default. Remove the checkmark from Hide System and select **Submit** to display System contributions.

Contribution Detail

The Contribution Detail page displays a selected contribution version. You can change the state of the contribution or delete the contribution.



Setting the State to Offline or Online

If the contribution state is online, it is the current version, and you can select **Set to Offline**. If there are other contribution versions, the next higher or lower version is set to online. If process instances are running against the BPEL files in the contribution, the contribution is set to Offline Pending. When all process instances complete, the state changes to Offline.

If the contribution state is offline, select **Set to Online**. The contribution's files become the current version. If any BPEL was deployed with an online (effective) date, the process state is online pending.

Note that the state of BPEL processes within a contribution is the same as the state of the contribution itself, but you can individually select a BPEL process from an online contribution and change its state. For example, if three processes are included in an online contribution, you can take one, two, or all three processes offline.

Deleting a Contribution

Select **Delete** to remove the entire unit of files from the catalog. If running processes exist, the contribution goes to the offline pending state until the processes complete.

When you delete a contribution, the following events occur:

- All files in the contribution are deleted.
- The deployment log is deleted.
- All completed active processes attached to the contribution are deleted.

Contribution Detail Information

The information within this page is grouped into sections, and these are described in the following list:

- **Contribution Detail**
  Each contribution is assigned a deployment Id. The name, state, version, and group details are the same as shown on the Contributions list page. A Description is available if it was provided during the BPR export in Process Developer.
- **Deployed Processes**
  List of BPEL files deployed, if any, and their current state. .
- **Contributed Resources**
  List of files deployed in the contribution. Examples of these files are WSDL, schema, HTML, XSL, XML, CSS, images, i18N properties files, Process Central files, event documents, Java jars, and report definitions. The version is the updated version associated with this contribution.

  WSDL and schema are always updated when they are deployed, even if identical to an earlier version. Other resource only have their version numbers incremented if their content has changed.

When you click on one of these links, Process Server displays its source, if possible. For example, it may display XML or HTML.

- **Exported Namespaces**
  Namespaces exposed by a resource in the contribution for use by other contributions.
- **Imported Namespaces**
  Namespaces from other contributions that this contribution is dependent on, if any. Note that if this contribution depends on another contribution, delete this contribution first.
- **Deployment Log**
  Link to the deployment log for this contribution.

Converting Legacy Deployments to Contributions

Process Developer Version 8.0 introduces contributions. A *contribution* is a deployed business process archive (BPR), managed as a unit of files. Rather than deploy and replace individual BPEL processes and resources, you deploy both current and updated as a unit that is easily managed on the server. For details, see *Managing_Deployment_Contributions*.

If you are upgrading to a new version of Process Developer, note that all BPR deployments prior to Version 8 are preserved as legacy contributions. On the Contributions page of the Process Console, you will see one or more entries named `legacy:/bpr.deployment` The newest deployment has the status of *online*. Previous deployments have the status of *offline pending*.

We highly recommend that you delete old BPRs and redeploy your processes and resources to take advantage of contribution features.

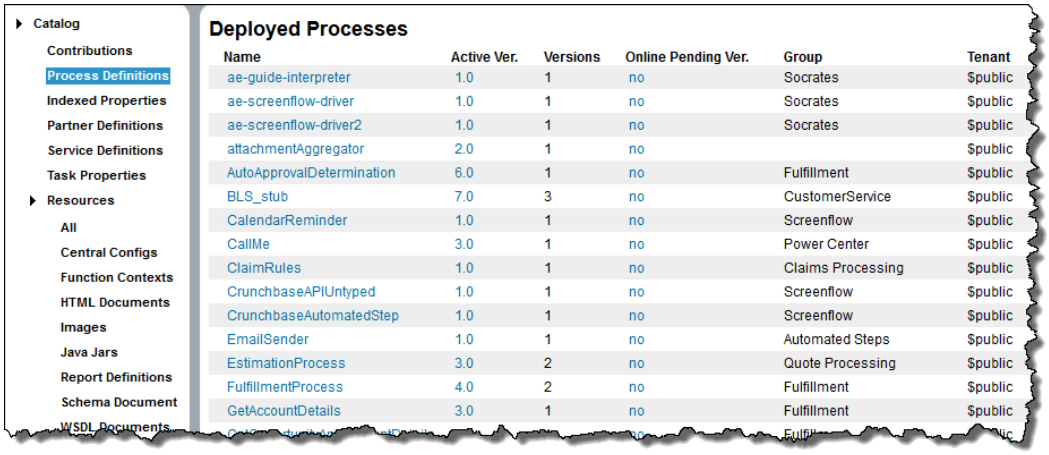
Here are some recommendations for converting your old deployments:

- Delete old BPRs and BPRDs from a project. You will create just one new BPR/BPRD per project. The BPR treats all deployable project files as a unit that is easily managed on the server.
- (Recommended). In your project, create a folder to hold all non-deployment files, such documentation files and unused resources. This is recommended, but is not required. You can add this folder to a list of excluded folders for deployment. See *Contribution Preference* for details.
- (Recommended). If your project is using WSDLs from another workspace project, you can specify that project as a project reference to manage those WSDLs in a separate contribution. For advantages to this approach, see *Using Project References*.
- Create and export a new project-based business process archive. Notice that the new Export Wizard automatically includes all deployable resources in your project and excludes test and sample data files.
- Be sure to deploy all deployable files each time you redeploy, even if you have only updated one or two files. Deploying an updated individual resource creates an undesirable version of a contribution.

Note that as you redeploy your BPEL processes and associated resources, they are removed from the `legacy:bpr.deployment` and added as contributions.

Viewing Process Definitions

The Process Definitions page lists all business processes that were deployed to the server. For each, basic process version information is displayed, as shown in the following figure and table.



Item	Description
Name	Local part of the process qualified name (`qname`)
Active Ver.	Version that process instances can attach to or can run to completion. Normally the active version is the current version. However, if the current version reached its expiration date, active processes can run to completion based on the offline pending (expired) version.
Versions	Number of deployed versions stored in the database
Online Pending Ver.	Yes-no field indicating whether a process version has an effective date set to a future date
Group	Group this process belongs to, as added in the PDD file.
Tenant	In a multitenant system, the tenant associated with the process definition.

Process versions can be in one of the following states:

- Online—By default, when no version information is specified in a deployment descriptor, a deployed process is the online (current) version with an immediate effective date. It is ready to receive requests.
- Online Pending—If an effective date is specified in a process' deployment descriptor, a process has an online pending (future) version.
- Offline Pending—A version is offline pending (expired) if reaches the expiration date specified in a process' deployment descriptor, you manually set the version offline and there are running process instances or a newer version is deployed while there are running process instances.
- Offline—When all process instances of an offline pending version complete, a process version is offline.

Select a process to display the **Deployed Process Version Detail** page. For additional information, see *Deployed Process Detail Graph* and *Process Version Life Cycles*.

Selection Filters

You can select the following Selection Filters to view a subset of processes:

- Process Status—As listed above:
- Process Name—Type in the name. You can use wildcards, such as tutor*.*, and select **Submit**.
- Process Group—Type in the group name, and select **Submit**. There is a default System group, which is hidden by default. Remove the checkmark from Hide System to display System processes.

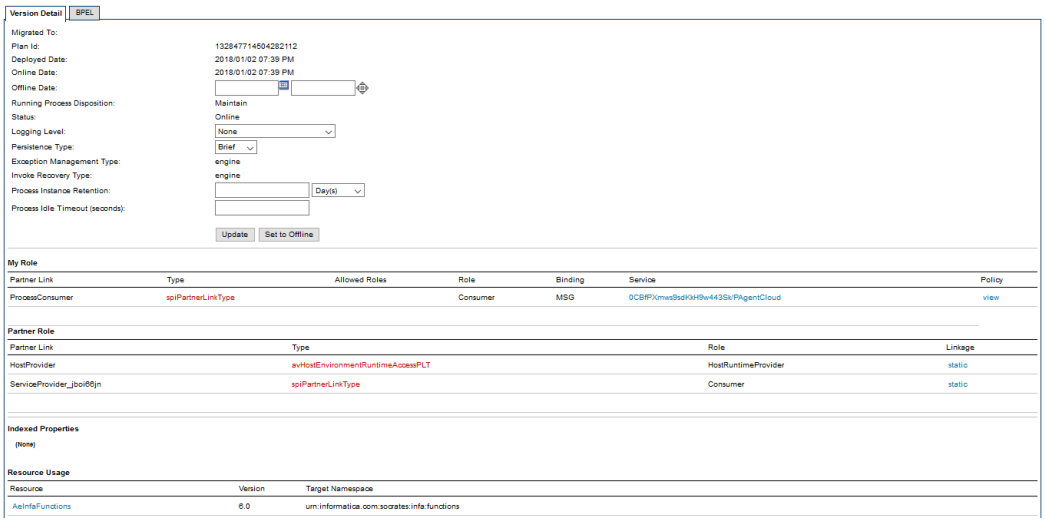
Deployed Process Detail Graph

The Deployed Process Detail Graph page presents many details about a process instance:

- An **Outline** view shows the structural elements of a BPEL process. You can select an element to view its properties and values.
- A **Graphic** view shows the main process flow. If the process has event handlers, fault handlers, and compensation handlers, you can view them by selecting a tab. You can also select an activity to view its properties.
- A **Properties** view appears for a selected element. For example, when the Process element is selected from the Outline view, you can see process properties and their current values.

Deployed Process Version Detail

The **Deployed Process Version Detail** page displays all the details from the process deployment descriptor as well as the process definition.



The sections of this page are described in the following topics:

- Process Version Life Cycles
- Updating a Process Version
- Setting a Process Version Offline or Online
- Logging Level
- Process Version Persistence Type
- Exception Management Type
- Invoke Recovery Type
- Process Instance Retention
- Deployed Process Detail Graph

The Deployed Process Version Detail page also shows endpoint reference details and other related details:

Role	Explanation
My Role	My Role partner link endpoint reference details are generated from information in the deployment descriptor. Rest your mouse on the partner link type to view the associated namespace. Select the Service Name link to view the WSDL file for the Web Service exposed by the partner link. Select View in the Policy column to view any attached policy assertions.
Partner Role	Partner Role partner link endpoint reference details are generated from information in the deployment descriptor. Rest your mouse on the partner link type to view the associated namespace. Select a static endpoint type from the Linkage column to view the endpoint definition.
Indexed Properties	Indexed Properties, if any, are displayed.
Event Filters	Event Filters, if any, are displayed. An event filter specifies process events that are passed to the Event Manager for processing. Event services, deployed to the Catalog, act upon the events.
Resource Usage	Resource Usage shows WSDL, schema, and other files and their target namespace referenced in this process. Select a Resource to view the resource definition.

Process Version Life Cycles

Process versioning allows different versions a process to exist in Process Server. Process versioning allows you to control when processes become effective and for how long. You can also control what happens to processes created by older versions when a new version becomes effective. While multiple versions of a process can exist concurrently, only the latest effective version can create new process instances.

The latest effective version is in an online state. Other states are:

- **Online pending**: describes versions that have an effective date in the future.
- **Offline pending**: describes versions whose expiration date has arrived, but running process instances are still active.
- **Offline**: describes expired versions that no longer have running process instances.

The process deployment descriptor (PDD) can contain a version element whose attributes describe how a deployment is versioned. These selections are all optional and have default values as described below.

The following example shows the syntax for version information in the `.pdd` file.

```
<version effectiveDate="2005-12-12T00:00:00-05:00"
         expirationDate="2007-12-12T00:00:00-05:00"
         id="1.5"
         runningProcessDisposition="migrate"/>
```

where:

- `effectiveDate` is the date the new version becomes the current version and all new process instances run against it.
  Depending on the disposition selected for running processes, some may continue to run until they finish using the older version. The effective date is an XML schema date/time value. The time expression includes a time zone, indicated as the midnight hour plus or minus the number of hours ahead of or behind Coordinated Universal Time (UTC) for the computer's time zone. In the example above, the computer time zone is Eastern Standard Time, which is five hours behind UTC. If you do not provide an effective date, it defaults to the date and time the process is deployed to the server.
- `expirationDate` is the date, beyond the online date, the current version expires.

An offline version is not capable of creating new process instances. Once all of the running processes tied to an offline pending version complete, the version becomes offline. All process instances for the current version run to completion. The expiration date is an XML schema date/time value. (Same as effective date). If you do not provide an expiration date, the version does not expire until you manually set it to offline in the Application Integration Console or until a newer version is deployed.

- `id` is the process version number in `major.minor` format.
  You do not need to provide a version number as Process Server auto-increments new versions. The server increments a version number by dropping the minor value and adding 1 to the max number. For example, version 1.5 increments to version 2.0.
- `runningProcessDisposition` is the action Process Server takes on any other versions of the same process that currently have processes executing after this version's effective date arrives.
  Values for `runningProcessDisposition` are:
  - `Maintain`. All process instances for the previous versions should run to completion. This is the default value.
  - `Migrate`. All running process instances created by previous versions will have their state information migrated to use the newly deployed process definition once its effective date arrives. If there are incompatible changes between the versions, descriptive warning messages are written to the Application Integration Console Server Log. Refer to the *Migrating Running Processes when Warnings are Generated* for details.
  - `Terminate`. Indicates that all process instances running under previous versions should terminate on the effective date of the new version, regardless of if the process instances are complete.

Updating a Process Version

You can view this property on the **Deployed Process Version Detail** page.

The options for updating a process version are:

- Depending on the version status (Online, Online Pending, Offline, Offline Pending), you may be able to update the online date, offline date, and running process disposition. For example, you can add an Offline Date to the online version, and select **Update**.
- You can also set the Process Instance Retention Days and update the process version. For details, see *Process Instance Retention*.

Setting a Process Version Offline or Online

You can view this property on the Deployed Process Version Detail.

The options for setting a process version offline or online are:

- To inactivate this version at a specified date and time, type in the values in the Offline Date fields.
- To inactivate this version immediately, select **Set to Offline**. No new process instances can be instantiated from this version. The status is Offline Pending until all running processes complete.
- If you had previously set this version offline, you may be able to restore it. To restore this version to the online or online pending (future) version, select the available option, **Restore to Online** or **Restore to Online Pending**.

Logging Level

You can set this property on the **Deployed Process Version Detail** page.

You can view or download an execution log for a running or completed process. An execution log provides start and end times for activity execution and helps you troubleshoot faulted processes. The following list describes the different logging levels:

- **None:** The Process Server does not log any information. Use this option to enhance engine performance.
- **Fault:** The Process Server logs only fault information. If no faults occur in a process, the Process Logging level defaults to **None**. Select **Fault** to reduce the size of the log file.
- **Execution:** This is the default option. The Process Server logs all execution statements except for *Will Not Execute* statements. Select **Execution** to decrease the size of the log file.
- **Execution with Data:** The Process Server logs all execution statements except for *Will Not Execute* statements, but includes variable, expression, and partner link data. Select **Execution with Data** to decrease the size of the log file.
- **Execution with Service Data:** The Process Server logs all execution and fault information, as well as some WSIO activity information. For execution information, Process Server logs deadpath states, terminations, ready-to-execute, and so on. For WSIO, Process Server logs invokes, picks, and receives, but excludes information related to data assignment or changes.
- **Full**: The Process Server logs all execution statements *Will Not Execute* statements for deadpath activities. For example, the Process Server logs all fault handling statements that are not executed.

To improve processing speed, perform no logging or minimum logging. Informatica recommends that you use the **None** or **Terse** logging levels.

Process Version Persistence Type

You can set this property on the Deployed Process Version Detail page.

Persistence refers to storage of active processes. By default, when a process runs on the server, all state and variable data is stored in the Process Server database. However, this setting can be changed in the PDD file to increase server performance and reduce database size.

Persistence setting selections are as follows:

Setting	Explanation
Full (default)	For each process instance, all running, faulted, and completed state information is stored. In the event of a server failure, a running process can be fully recovered. The recovery is possible because this setting tells Process Server to maintain a journal (a record of the changes intended for the database). **Note**: If the process uses a WS-RM invoke handler for a partner role or a WS-Reliable Messaging policy assertion on a my role, full persistence is required.
Persist	Same storage setting as Full, but without journaling. If processes are running and the server fails, processes are suspended. The process is recoverable if the system goes down, but needs to be looked at since no journaling was done. This process is marked as suspended.
Final	Stores only the final state of the process (completed or faulted) and process variables. On a server failure, a running process is terminated. This setting makes fewer database writes than the previous two settings, but still allows you to view a graph of the process on the Active Processes Detail page in the Application Integration Console. Here, you can see the execution path and final values of process variables. A process runs only in memory, and the Server Property called Process Idle Timeout has no effect on this persistence level.

Setting	Explanation
Brief	This is the minimum level for process logging. Process Server stores only the start and completion times and the final state (completed or faulted). Also, it stores state and process variables only if the process faults. A process runs only in memory, and the Server Property called Process Idle Timeout has no effect on this persistence level.
None	No process information is stored in the server database when a process terminates. The process instance is not listed in the Processes page.

Exception Management Type

You can view this property on the **Deployed Process Version Detail** page.

According to the WS-BPEL 2.0 specification, a process with an uncaught fault terminates.

On the Server Properties page, you can enable an option to suspend all processes on an uncaught fault to put them in a suspended-faulting state. You can then perform process exception management on the faulting process, followed by retrying or completing the faulting activity or scope.

An individual process can override the engine setting with an entry in the Process Deployment Descriptor (PDD) file. The settings are:

Field	Explanation
System Default	The current engine setting for all processes. The default engine setting is to disable suspension on uncaught fault; however, the current setting may be different.
False	Do not allow this process to suspend on an uncaught fault. The process will terminate abnormally. This setting overrides the engine setting.
True	Suspend this process on an uncaught fault to put it in a suspended-faulting state. You can then perform process exception management on the faulting process, followed by retrying or completing the faulting activity or scope. This setting overrides the engine setting.

See *Process Exception Management*.

Invoke Recovery Type

You can view this property on the Deployed Process Version Detail page.

For invoke activities that do not complete because of a node failure, you can suspend the process upon recovery. The process is suspended at the pending invoke, and you can perform process exception management.

An individual process can override the engine setting with an entry in the Process Deployment Descriptor (PDD) file. The settings are:

Setting	Explanation
System Default	The current engine setting for all processes. The default engine setting is to disable invoke recovery; however, the current setting may be different.
False	Do not allow a pending invoke activity to suspend upon recovery. The process will terminate abnormally. This setting overrides the engine setting.
True	Suspend a pending invoke on recovery to put it in a suspended-faulting state. You can then perform process exception management on the invoke activity, followed by retrying or completing the faulting activity or scope. This setting overrides the engine setting.

Process Instance Retention

You can view the **Process Instance Retention** property on the Deployed Process Version detail page.

You can specify how long to keep completed and faulted processes in the Process Server database before deleting them on an automated schedule. This setting is available in the following locations:

- Process Deployment Descriptor editor in Process Developer. You can add the setting to a process's deployment descriptor.
- The Deployed Process Version detail page of the Application Integration Console. Add or change the setting for a deployed process.
- The default retention days setting for all processes on the Application Integration Console. The setting for an individual process overrides this setting.

The retention setting applies to a process version with a status of online or online pending, and not to offline pending or offline versions. The schedule begins on the completed date or faulted date of each process, as shown on the **Processes** page. For example, if a process instance completes on December 31, and the retention setting is 30 days, the process instance is deleted from the database on the next scheduled deletion after January 30.

Configuring Retention for an Individual Process

1. In the Application Integration Console, select **Deployed Assets** > **Deployed Processes**.
2. Select a process from the list.
3. If multiple versions exist, from the **Deployed Process Detail** page, select an online or online pending version to open the **Deployed Process Version Detail** page.
4. In the **Process Instance Retention** field, specify the number of retention days, hours, or minutes for the process version.
5. Select **Update**. The change takes effect immediately for all completed and faulted processes.

For details about scheduled database maintenance, see *Storage*.

Deployed Process Detail Graph

The Deployed Process Detail Graph page presents many details about a process instance:

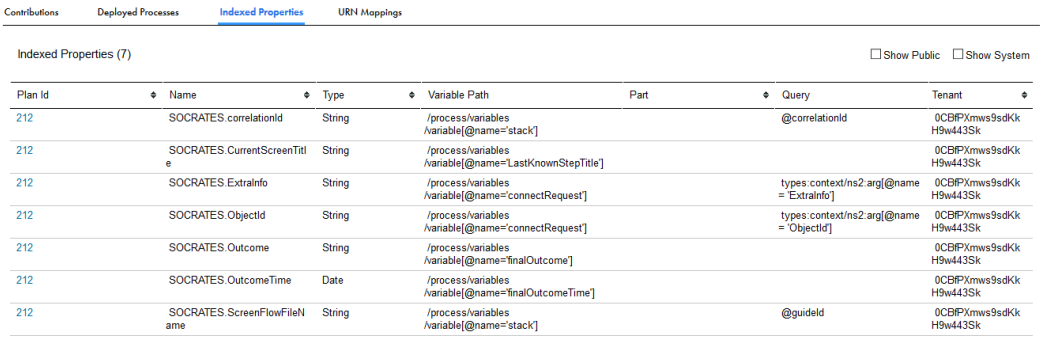
- An **Outline** view shows the structural elements of a BPEL process. You can select an element to view its properties and values.

- A **Graphic** view shows the main process flow. If the process has event handlers, fault handlers, and compensation handlers, you can view them by selecting a tab. You can also select an activity to view its properties.
- A **Properties** view appears for a selected element. For example, when the Process element is selected from the Outline view, you can see process properties and their current values.

Indexed Properties

An indexed property is a variable property that serves as a selection filter for active processes. It can also be used in defining events for business event processing. This property holds a piece of data, such as a customer Id, application date, or amount. Using an indexed property in an event service allows for fine-grained analysis of business events as they are monitored by the Event Manager. Using an indexed property in a selection query provides a fast way to filter processes based on important data items.

The following image shows a sample **Index Properties** tab .



Contributions | Deployed Processes | Indexed Properties | URN Mappings

Indexed Properties (7) ☐ Show Public ☐ Show System

Plan Id	Name	Type	Variable Path	Part	Query	Tenant
212	SOCRATES.correlationId	String	/process/variables /variable[@name='stack']		@correlationId	0CBfPXmws9sdKk H9w443Sk
212	SOCRATES.CurrentScreenTitle	String	/process/variables /variable[@name='LastKnownStepTitle']			0CBfPXmws9sdKk H9w443Sk
212	SOCRATES.ExtraInfo	String	/process/variables /variable[@name='connectRequest']		types:context/ns2:arg[@name = 'ExtraInfo']	0CBfPXmws9sdKk H9w443Sk
212	SOCRATES.ObjectId	String	/process/variables /variable[@name='connectRequest']		types:context/ns2:arg[@name = 'ObjectId']	0CBfPXmws9sdKk H9w443Sk
212	SOCRATES.Outcome	String	/process/variables /variable[@name='finalOutcome']			0CBfPXmws9sdKk H9w443Sk
212	SOCRATES.OutcomeTime	Date	/process/variables /variable[@name='finalOutcomeTime']			0CBfPXmws9sdKk H9w443Sk
212	SOCRATES.ScreenFlowFileName	String	/process/variables /variable[@name='stack']		@guideId	0CBfPXmws9sdKk H9w443Sk

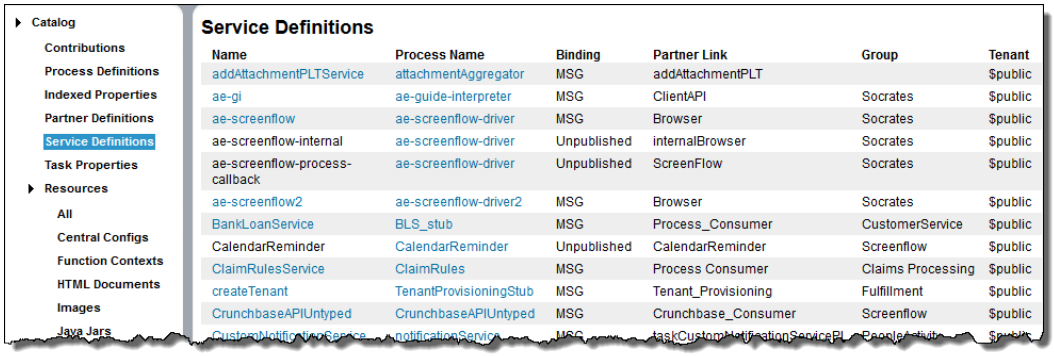
For example, you can retrieve a list of faulting processes that share the same indexed property, suspend processes, fix bad data values, and continue process execution.

Indexed properties are defined in the process deployment descriptor file. Information in this table is as follows:

Item	Description
Plan ID	The deployed process associated with the indexed property; clicking on this link displays the Deployed Process Version Detail Page.
Name	Indexed property name. This name appears in the Indexed Property list in the selection filters Expression Builder.
Type	Schema type, such as string or double
Variable Path	Process variable name and declaration location in the process
Part	Process variable part for message type variables
Query	Process variable part detail (optional)

Service Definitions

A deployed process contains at least one My Role partner link, and a My Role partner link is assigned a service name in the Process Deployment Descriptor (PDD) file. The service name identifies the WSDL that the Process Server engine generates during deployment. The WSDL includes the messages, operations, service, and binding details for the Web Service exposed by the process' My Role partner link.



The process receives messages at the Web Service address, which is shown in the following example:

```
http://localhost:8080/active-bpel/services/[servicename]
```

In addition to this default service endpoint, Process Server provides several others, including REST, JSON, and XML. For details, see the *Process Server* topics of this help.

Note: Some services are deployed as unpublished, indicating they are not exposed as Web Services. An example is a Retry Policy system service, which is a process deployed to tell the engine how many times to retry a non-communicating service. This type of process is not intended for outside consumption. For more information on unpublished services, see the *Process Developer* Help.

The following details and links are included on the Service Definitions page.

Field	Explanation
Name	Service name assigned to a My Role partner link in the PDD file. Select the name to link to the WSDL generated for this partner link. The WSDL is the Web Service that receives inbound messages. The WSDL address is in the form of: `http://localhost:8080/active-bpel/services/[servicename]?wsdl` Clicking on this link shows the services source.
Process Name	Process containing the My Role partner link associated with this service. Clicking on this link displays the **Deployed Process Detail** page .
Binding	Standard SOAP binding styles indicating how to format inbound messages for the service. Values are: - MSG (Document Literal) - RPC Literal - RPC Encoded - Unpublished (system service) - Policy
Partner Link	Name assigned to a My Role partner link that is exposing the service. Select the name to link to the **Deployed Process Version Detail** page.

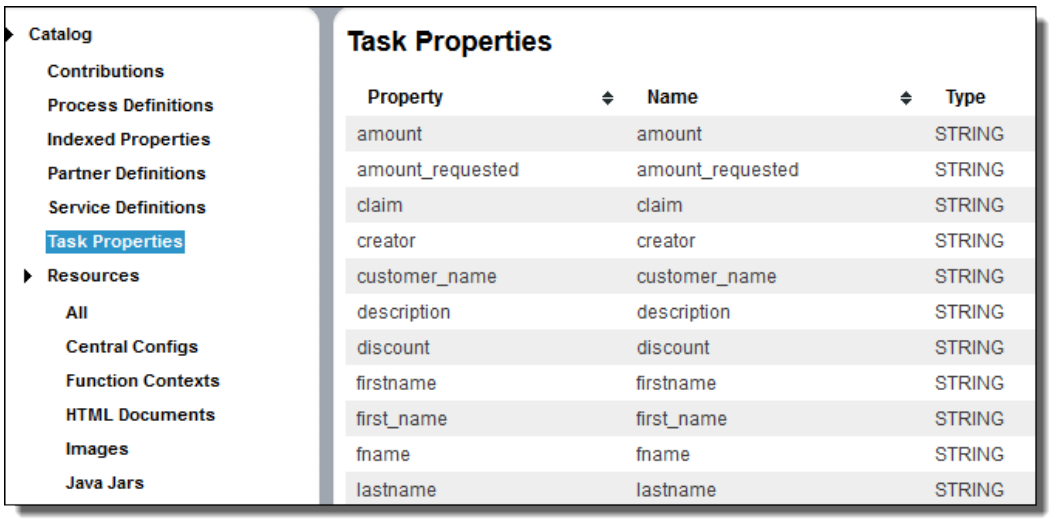
Field	Explanation
Group	The name of the group representing users and other groups. See for more information.
Tenant	The tenant's name. See .

Using the Selection Filter

To display a service, you can type its name into the Service Name field and select **Submit**. You can also use the asterisk (*) wildcard to search for names. For example `*par*` returns all service names containing the “par” characters. To view all services, leave the Service Name field blank and select **Submit**.

Task Properties

A task property is a well-known WS-HT API property or a user-defined property that serves as a selection filter in Process Central and other task clients.



The list of task properties includes only user-defined properties. A developer defines a property as a presentation parameter in a Human task in Process Developer.

The list shows all properties created from all processes. In Process Central, a property can be displayed as a custom column in a task list and can also be used as a search criteria. If used as a custom column, the task column definition is included in an `.avcconfig` file.

If a property is defined by more than one data types, the property name appears for each type.

Resources

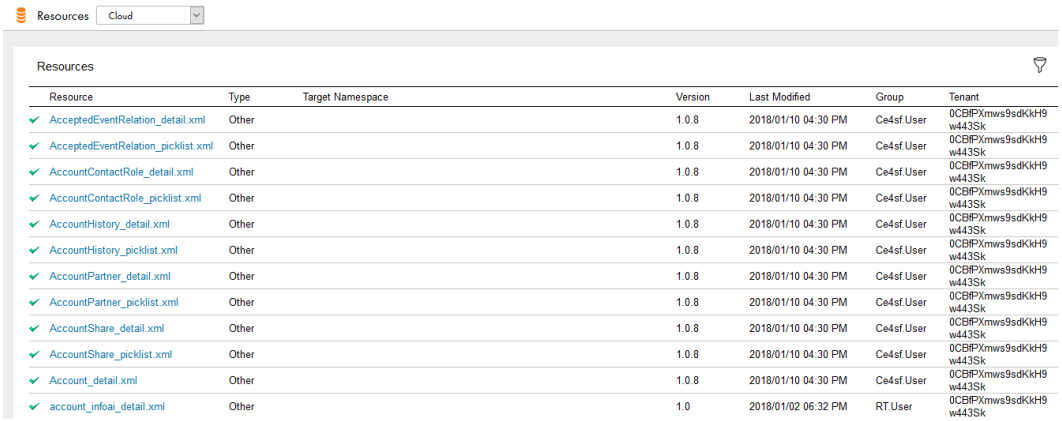
The **Resources** page contains the following resource types:

Resource Type	Description
Central Configs (on-premises only)	Configuration files governing the display of folders and other filters for Process Central Tasks, Forms, and Reports.
Function Contexts	POJO custom functions imported into processes.
HTML Documents	Forms for initiating processes as well as task forms for People activities. Used in Process Central.
Images	Image source files referenced in HTML pages.
Java Jars (on-premises only)	Interfaces for partner services.
Report Definitions (on-premises only)	System and user-defined reports.
Schema Documents	XSDs imported into processes.
WSDL Documents	WSDLs imported into processes.
XQuery Modules	XQuery custom functions imported into processes.
XSL Documents	Transformational style sheets for custom functions and the older Inbox application renderings.
Others	- XML files for custom functions, sample data, numerous other configuration files - .properties files for multilingual support - JavaScript files for forms

Viewing Catalog Resources

Any resource can be accessed by any deployed BPEL process and only one copy is maintained.

The following image shows the Resources page:



When you view the **Resources** page for a Secure Agent, it displays the following details on how the node is accessing the catalog (the details do not display if you view the Resources for the Cloud Server):

Item	Description
Total Reads	The number of reads to retrieve resource information during process execution (in cache or not).
Disk Reads	The number of reads made to resource files not in the cache expressed as an absolute number and percentage of Total Reads.
Cache Efficiency	The percentage of resources that are read from memory versus read from storage. For example, 100% indicates that all resource reads are coming from the memory cache.
Cache Size	The number of resource files in stored cache. The default is 100. You can set cache size on the **Admin** > **Configure Server** > **Server Properties** . Modifying the cache size may improve engine performance.

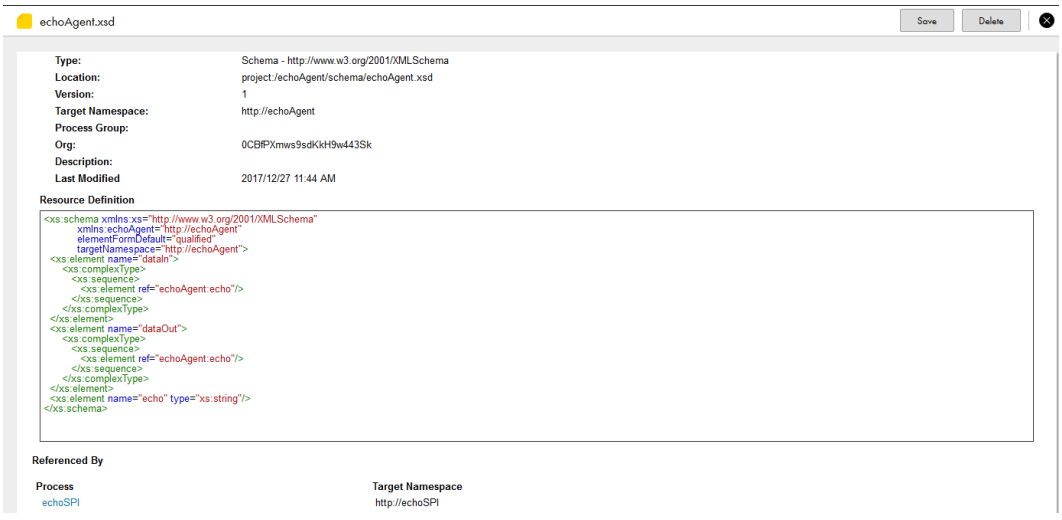
The Resources list shows the type, name, and namespace for the resource. Rest your mouse on the Resource name to view the physical location where the resource was loaded from.

The information about a resource on this page is the same as that is displayed in the Resource Detail page. Clicking on a resource link displays the Resource Detail page.

Resource Detail Page

The **Resource Detail** page shows the same information that is on the Resources page for each resource. It also displays the XML source code.

The following image shows a sample Resource Detail page:



Property	Description
Name	The name of the resource
Class	The class from which the resource was instantiated
Location	The physical location from where the resource is loaded. This helps to uniquely define the location when the deployment descriptor was created and can be used to have multiple resource files of the same name deployed to the engine. The WSDL location is referenced in the `.pdd` file.

Property	Description
Version	The version of the resource. This is either automatically created by the Process Server or assigned to in the `.pdd` file.
Target Namespace	Target namespace in the resource
Process Group	Group name that was added in the `.pdd` file; this is used to organize resources
Description	Description that was added in the `.pdd` file
Last Modified	The date at which this resource was last changed

The **Resource Detail** page shows the process versions referencing this resource for some resources.

Selection Filters

Use Selection Filters as follows:

- Select a State—All, Online or Offline. For details, see *Contributions*.
- Select a resource Type from the list.
- Type in a Resource name—You can use wildcards, such as `tutor*.*`
- Type in a Target Namespace—You can use wildcards.
- Type in a Group name—If a group name was defined during deployment, you can search for it.

Select **Submit** to display the filtered list of resources.

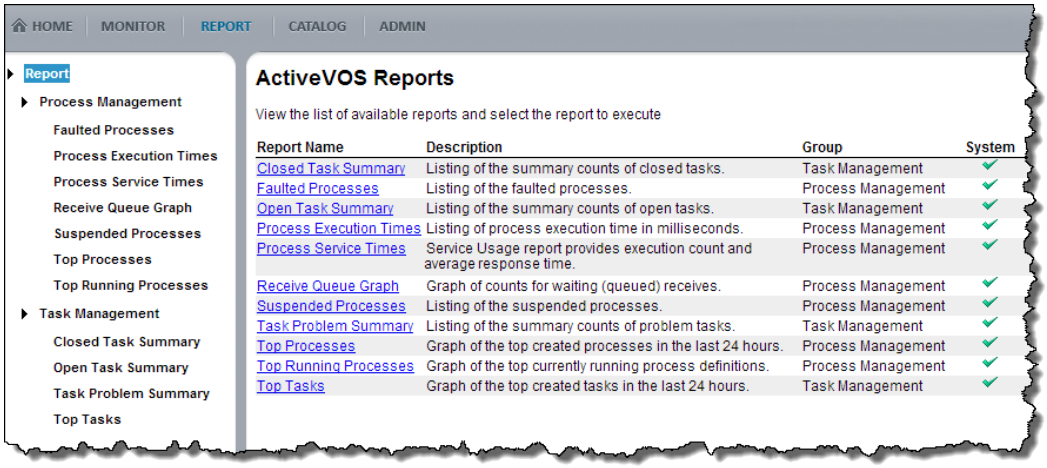
Reports (Process Developer)

The Dashboard and Report pages display business activity monitoring reports. These pages help you understand statistics about active processes. Process Server includes several basic reports, and you can deploy your own custom reports to the server.

The Dashboard gives you a quick status of where your system is by displaying thumbnail reports of the most frequently needed information. You can click on a thumbnail to link to a larger display of the report on the Reports page.

The Reports page has two main categories of reports:

- **Process Management**: Process-based reports show data about any active process
- **Task Management**: Task-based reports show data about active human tasks



An *active* process or task is one that has started, and perhaps completed, execution. A report may provide greater state detail about running, faulted, or suspended processes and similar detail about unclaimed, started, or suspended tasks.

Reports have the following features:

- A report consists of data retrieved from the database at an interval basis set internally. The date and time of the report is displayed.
- You can select a link on a report item to view details about the active process or human task.
- Toolbar icons on a displayed report allow you to open and save a report to applications such as Microsoft Word or Excel.
- Top processes and tasks (for the Top reports) are the most active, based on a 24-hour rolling window

Report Viewer Toolbar

Use the report viewer toolbar as follows:

- **Toggle table of contents**. For user-defined reports that define table of contents entries, you can turn the display on or off. The basic Process Server reports do not use a table of contents.
- **Run report**. For user-defined reports that include runtime parameters, you can select parameters to display different report data. The basic Process Server reports do not use runtime parameters.
- **Export data**. You can select data elements from a report and then select export preferences for the data.
- **Export report**. Export a report in a format compatible with applications such as Microsoft Word or Excel.
- **Print report**. Print a report in HTML or PDF format.

User-Defined Reports

You can create your own reports in Process Developer and deploy them to the server. During deployment, you can specify a Process group name or Contribution group name for a report category. Using a group name, your reports can be grouped like the process and task reports on the Reports page.

Process Developer provides a project template that contains the configuration details needed for creating reports for deployment to the server. The template contains a connection to database tables and views as well as style elements for the layout and display of reports.

A comprehensive report creation environment lets you create text or graphical reports.

For details, see the *Creating Reports for the Process Server* in *Process Developer Help*.

ActiveVOS Custom Faults

The following list specifies the custom faults defined for the ActiveVOS engine. All these faults are in the namespace

http://www.active-endpoints.com/2004/06/bpel/extensions/

Fault name	Fault name
`systemError`	Unrecoverable system error
`badProcess`	Invalid BPEL
`validationError`	Error in message variable data. Validation errors are reported only if the configuration option “Validate input/output messages against schema” is enabled.
`xpathFunctionError`	Error in executing XPath function
`invalidTransitionCondition`	Non-Boolean return from an XPath evaluation of a transition condition
`xpathDateParseError`	error in parsing an xsd:date or xsd:datetime
`xpathDurationFormatError`	error in parsing an xsd:duration

CHAPTER 5

Admin

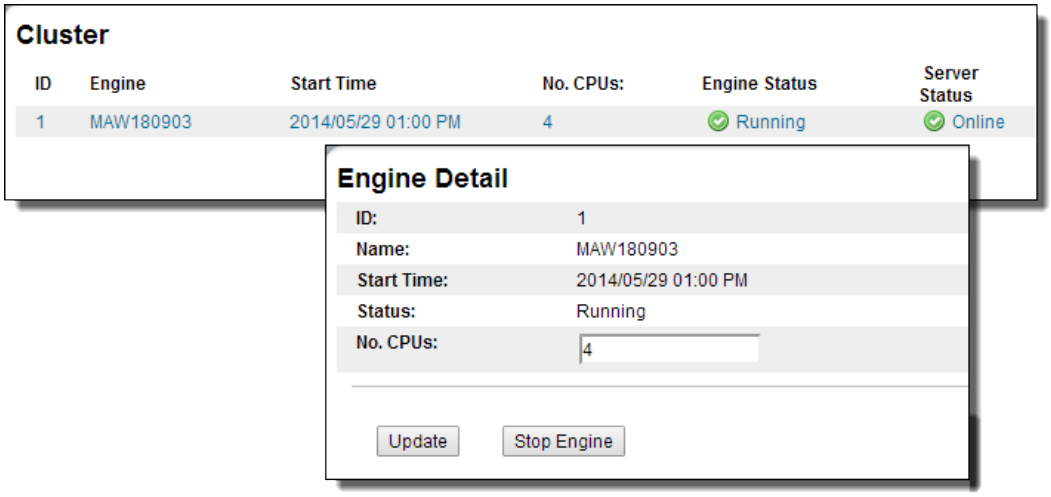
This chapter includes the following topics:

Cluster

Process Server supports clusters of Process Servers operating with a single persistent process store. Multiple process servers share a set of process instances and their deployments.

Two main features of a clustered environment are load balancing and fail over. Load balancing allows processes to be routed to the server in the process cluster with the least load. Fail over allows servers to pick up the load of a failed server in the process cluster.

The following figure shows the Cluster page and the Detail page that is displayed if you click on a link in the Cluster page.



Clicking either of an engine's links displays the Engine Detail page, which has the following information:

Field	Explanation
ID	ID assigned to the cluster node by the Process Server engine
Name	Computer IP address and port where the engine instance is running
Start Time	Date and time the engine started
Status	Statuses include Running and Stopped
No. CPUs	Number that is based on licenses. Select the number of CPUs to view the Engine Detail page, where you can update the number. Then view the **Admin** > **License** page to verify that the number of licenses matches the number of CPUs. Process Server reports an estimated number of CPUs based on the information it gathers from the computer. The number may not be accurate if the computer is using CPU hyper-threading technology.

Stopping this Engine

Press the **Stop Engine** button to stop this engine. To stop all engines in the cluster, stop the cluster on the Home page. For details, see *Server Status*

Updating the CPU Count

1. Click on a link within the Cluster page for the engine being updated.
2. Stop the engine.
3. Update the number of CPUs in your cluster.
4. Select **Update**.
5. Restart the engine.

WebSphere Cluster Properties

If you are using WebSphere, use the Cluster Properties page to set WebSphere Cluster properties. (Other engines will show different properties.)

The following are WebSphere cluster properties:

- **Cleanup Delay (seconds)**. This delay is the amount of time a node waits on startup before attempting to recover work from another engine. The default is 300 seconds.
  You can increase this value if you notice that the startup of a cluster node is particularly slow. The slowness may be due to contention between nodes for recovering work. A larger value gives a node a chance to startup before another node contends for its recovery work.
- **Time Difference Tolerance Warning (seconds)**: The difference in seconds that nodes within a cluster can be out-of-sync with one another before a warning state is set. This lets you detect if the time setting is not in sync between cluster members.
- **Failover Delay (seconds)**. This delay is the amount of time Process Server waits before triggering failover after a node leaves the cluster. (*Failover* means that work that was in process when the node failed needs to be completed by another server that is currently online.)
  After the timeout, Process Server confirms that the node did not rejoin the cluster and still needs failover. Additionally, Process Server issues a warning regarding a likely glitch in cluster communications if a node leaves and rejoins the cluster but the engine itself remained running the whole time. The default delay is 30 seconds.
- **Automatic Failover Enabled**: When communications to a node are lost, failover occurs automatically.
- **Log Inbound Cluster Communication**: The server logs all inbound cluster calls, along with relevant data it receives. This information is extremely useful when you are debugging cluster configurations issues.
- **Log Outbound Cluster Communication**: The server logs all outbound cluster calls, along with relevant data being sent to nodes in cluster. This information is extremely useful when you are debugging cluster configurations issues.
- **Membership Update Interval**: Indicates how often you want to poll engines in the cluster to ensure they are active. The default value is 30 seconds.

You can add the name of the WebSphere application server JAAS login for cluster communications.

If your WebSphere application server is running with global security enabled, you can provide or update the JAAS username. Process Server uses this name when performing cluster communications. You can choose from the following:

- **ActiveVOSProvidedUser**. JAAS user created with Monitor rights
- **ActiveVOSIdentityAssertion**. JAAS user created with Monitor rights and uses the user name and password of the requester to reassert the credentials
- **JAAS Custom Login**. JAAS user that is available on the WebSphere application server that has at least Monitor rights

In WebSphere, you can also set a Communications Time Out value for Process Server to send and receive communications to and from engines in the cluster. The default value is 180 seconds.

Weblogic Properties

You can set a Membership Update Interval to indicate how often you want to poll engines in the cluster to ensure they are active. The default value is 30 seconds.

License

The License page displays the installed licenses for your Process Server product. You can re-install, add, and remove licenses as needed.

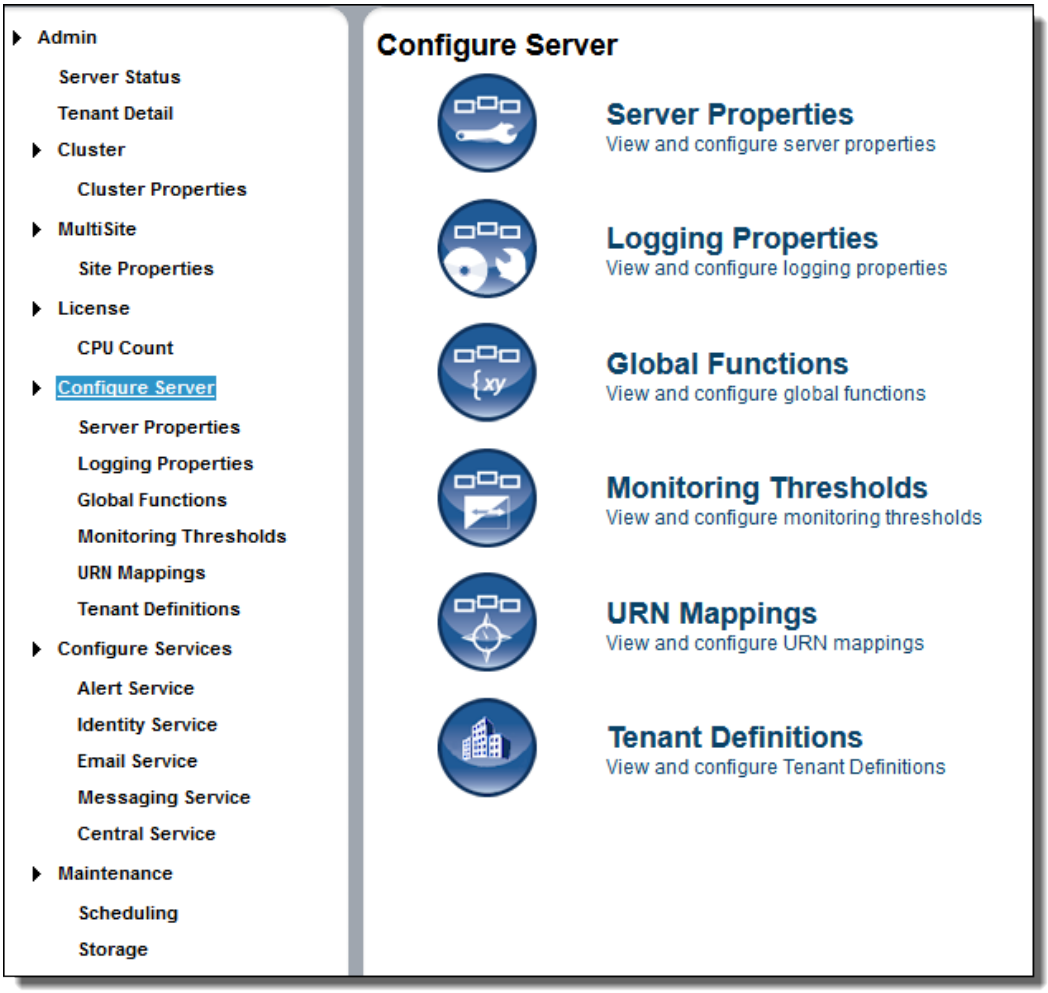
Note that you must have at least one license installed in order to run the engine.

Do the following on this page:

- To add or re-install a license, select **Add License**. In the dialog box that opens, paste in the license key that was emailed from Informatica. Select the **Add License** button in the window.
- To remove a license, select **Remove** for one of the installed licenses
- To view details for one license, select a serial number under Installed Licenses. The License Detail page shows specifics for the organization, application server, license type, and number of licensed CPUs. This information is similar to what is displayed here.

Configure Server

After selecting the Configure Server command, Process Server displays the following page:



Configure Server topics include:

- Server Properties: View and configure server properties
- Logging Properties: View and configure logging properties
- Global Functions: View and configure global functions
- Monitoring Thresholds: View and configure monitoring thresholds
- URN Mappings: View and configure URN mappings
- Tenant Definitions: View and configure Tenant Definitions

Server Properties

To see Process Server, select a Secure Agent.

On the Server Settings tab, you can make configuration changes without stopping and restarting the engine. When you change the properties and select Update, the changes take effect immediately. The changes are also written to the database and propagated to other engines in the cluster.

View and update Process Server configuration settings as shown in this topic. Note that some of these properties are the same as the Process Developer Simulation preferences.

Allow proxy user in process invocations

A process can have a policy assertion (called Run as User) that enables a named user to be a process initiator, as opposed to the normal process initiator with credentials or an anonymous user. If desired, you can disable this property and the policy assertion is ignored. This setting applies to all new process instances and is enabled by default.

Auto create target path for Copy/To

Applies only to processes that are validated against the BPEL4WS 1.1 specification. For WS-BPEL 2.0 processes, this property can be added as an extension on a per process basis. Refer to the *Process Developer Guide* for more information.

This property determines if Process Server can create a location path for a non-existent node in a complex variable in a process instance document. When an assignment refers to a non-existent node (or to more than one node), the standard BPEL fault, `bpws:selectionFailure`, must be thrown, according to the BPEL specification.

Enabling this option allows selections to be created on-the-fly. This means an assign copy TO operation can refer to a non-existent node and assign a value to it. This option is disabled by default.

Contribution Cache

Setting this cache controls how many contributions are kept in memory. This cache avoids having to read the database too often. Generally contributions are small. However, the size depends on how many resources (that is, catalog entries) they contain and how many imports and exports they have.

Changing the default value depends on the number of deployed contributions and the amount of memory available. Default is 100 contributions. If a contribution is cached, it need not be looked up in the database. If you have more than 100, you may want to increase this value.

This cache setting doesn't require much tuning and profiling the application server is probably required to determine if changing this value would improve performance.

This cache is not set in the **Server Properties** dialog. Instead, you will need to set this in the engine configuration file, which is `aeEngineConfig.xml`, which is contained in the `activvos.war` file.

Deployment Lock Timeout (seconds)

In a clustered environment, one machine acquires a deployment lock, and the other machines must wait until deployments are complete before starting `.bpr` deployments. If you have many machines in a cluster, you may need to increase the timeout value because each machine deploys many system `.bprs` upon start-up. Default is 120 seconds.

Deployment Plan Cache

A deployment plan corresponds to each deployed version of a process, including associated disposition of running processes. Process versions that are active can be cached for better engine performance. Default number of plans that are cached is 100. For details regarding versions, see *Process Version Life Cycles*.

Disable bpws:selectionFailure fault

Applies only to processes that are validated against the BPEL4WS 1.1 specification. For WS-BPEL 2.0 processes, this property can be added as an extension on a per process basis. Refer to the *Process Developer Guide* for more information.

This option allows an XPath query in the FROM clause of an assignment statement to return an empty node set. If it does, the target node for the assignment is deleted.

By default, this option is not enabled, and if the query string returns an empty selection from an assign copy FROM, the process throws a `bpws:selectionFailure` fault, which is the standard response described in the BPEL4WS specification.

Maximum HTTP Connections/Maximum HTTP Connections Per Host

If you have several HTTP Invokes within a process, the Invokes could spawn more than 100 connections. If this occurs, you should, you should increase the default maximum number of HTTP Connections value so that multiple threads are not suspended waiting to create a HTTP Connection. (You will see service call timeouts when this happens.)

Changes you make do not occur until you restart the server.

Message size limit

Sets the maximum size of the payload for a multi-part message that does not have attachments.

If you use the Informatica Cloud Server, you can set a maximum message payload size of 5 MB.

If you use a Secure Agent, the default value is 5 MB, but you can change set a higher or lower value.

Message TTL (seconds)

Specifies the amount of time between the time a message reaches the Process Server and the time when it is about to get dispatched to the target process. Default is 24 hours.

When you configure the message time to live (TTL) period and invoke a process by using the REST or SOAP endpoint, if the HTTP message is not dispatched within the message TTL period, the request fails. You see the HTTP error `503 Service Unavailable` with the following error message:

```
Message discarded having exceeded the defined TTL.
```

The message TTL is applicable to the endpoints with the following entries:

- `/active-bpel/services/`
- `/active-bpel/public/rt/`
- `/active-bpel/rt/`
- `/active-bpel/public/soap/`
- `/active-bpel/soap/`
- `/active-bpel/odata/repository/v4/OdataRepository/Execute`
- `/active-bpel/tf/`
- `/activebpel-cert/services/`
- `/activebpel-cert/rt/`
- `/activebpel-cert/soap/`
- `/activebpel-cert/odata/repository/v4/OdataRepository/Execute`
- `/process-engine/rt/`
- `/process-engine/public/rt/`
- `/process-engine/services/`
- `/process-engine/public/soap/`
- `/process-engine/soap/`

Message with attachments size limit

Sets the maximum size of the payload for a multi-part message that has attachments.

If you use the Informatica Cloud Server, you can set a maximum message payload size of 5 MB. The maximum size for attachments is also 5 MB.

If you use a Secure Agent, the default message payload size is 5 MB and the default attachment size is 5 MB. You can change the default sizes to set higher or lower values.

Migrate running processes even on warning

If Migrate is selected as a deployment option on a contribution, running processes are migrated to the new process version. The server checks for compatibility between the old and new process plans and generates a warning in the Process Server Log if an incompatibility exists.

Be sure that the Process Server Log is set to at least Warning Level before any migration is attempted.

If this property is disabled (the default), and if a warning condition exists for one or more running process instances, a copy of each running process instance is created, and both the pre-migration copy and the new copy are suspended. Additionally, a warning code/message is added to the Process Server Log.

If this property is enabled, no copy of the running instance is made if a warning condition exists. The only action is that a warning code/message is added to the Process Server Log.

For details, see *Migrating Running Processes when Warnings are Generated* .

Process Count

Specifies the maximum number of processes in memory. Default is 250. Specifying 0 indicates no limit, but this is not recommended.

Process Idle Timeout

Specifies the number of seconds to wait until process state information is written to the database during idle processing times such as waiting for a reply from an invoked service. You can increase the timeout value to enhance engine performance. You can decrease the value to ensure the full process state is always in the database. Doing so avoids potential process recovery time in the event of a server failure. Default is 10 seconds.

Resource Cache

The number of WSDL files and other resources in stored cache. Default is 100. Modifying the cache size may improve engine performance. A value of -1 means unlimited caching, but this is not recommended.

Screen Cache Size (MB)

Sets the size of the cache Process Server uses for storing screens used by guides. Guides can be large in size when they are published and generally not every single path in the guide is always executed. This cache holds the HTML for the most commonly used screens within a guide. This improves the time required to load a screen as the guide containing the screen need not be loaded.

Suspend process on invoke recovery

For invoke activities which do not complete due to the node failure, you can suspend the process upon recovery. The process is suspended at the pending invoke, and you can perform process exception management, if desired.

An individual process can override this setting with an entry in the PDD file.

Suspend process on uncaught fault

According to the WS-BPEL 2.0 specification, a process with an uncaught fault terminates.

Enable this option to suspend all processes on an uncaught fault to put them in a suspended-faulting state. You can then perform process exception management on the faulting process such as retrying or completing the faulting activity or scope.

An individual process can override this setting with an entry in the PDD file. See *Exception Management Type*.

See *Process Exception Management* .

Suspended processes termination batch size

Set the maximum number of suspended processes to terminate in a single run. You can set the batch size to control the overload on the system work manager thread pool.

Default is -1. A value of -1 means that is no limit set for terminating suspended processes, and all the suspended processes will be terminated.

Suspended processes termination timer interval (minutes)

Set the time interval in minutes at which the suspended processes must be terminated.

Default is 1440 minutes. This means that after 24 hours, the suspended processes start terminating.

Terminate suspended process after (days)

Suspended processes stay in the Active Processes list indefinitely, if they are not otherwise resolved.

Enter the number of days to retain suspended processes, for example, 30 days. The suspended processes are terminated automatically after the retention period is reached.

Default is 0. This means that the suspended processes are never terminated. For processes running on the Cloud Server, this option is determined by Informatica.

Unmatched Correlated Receive Timeout

Sets the number of seconds that the Process Server waits to match a correlated message to a message activity or an event activity if the message arrives before the activity becomes active. Set this value to avoid many unconsumed messages on the server.

Default is 30 seconds. If you enter an Unmatched Correlated Receive Timeout of 0 seconds, the Process Server discards all unmatched correlated messages when they arrive.

If the Process Server crashes, it recovers messages and continues to wait until a relevant consumer consumes the messages, or until a timeout occurs.

If the Process Server waits for a time that is longer than the value you specify, a timeout occurs. When a time out occurs, the Process Server adds a correlation violation error to the server log. This correlation violation error does not contain any message details. The correlation violation error contains a unique hash key called `AeSecuredLogDatakey`, a concatenation of the Engine ID, Plan ID, Queue ID, and Timestamp in milliseconds of when error occurred.

The following is a sample correlation violation error:

```
correlationViolation [AeSecuredLogDatakey=786000_123456_1000_1476434734123;tenant=ABCDEF;
planId=123456;partnerLink=MessageEvent1EventPL;operation=Initiate]
```

To view message details such as the service name, operation name, and message parts, search for `AeSecuredLogDatakey` in the `AeSecuredLogData` table.

The correlation violation error contains the hash key and not the message details because the message details might contain secure information. This two-step method secures your data.

Validate input/output messages against schema

Validates the data used in service interactions against their associated schema.

Enable this option to validate data before execution starts. Disable this option for faster execution. This option is enabled by default.

Web Service Invoke/Reply Timeout

For performance reasons, a reply activity matching a receive, as well as synchronous invokes, are timed out if they do not execute within 10 minutes. If you are receiving timeout errors, you can specify a greater amount of time to wait before a process is timed out due to a reply or synchronous invoke activity not executing within 10 minutes.

Default is 600 seconds.

Work Manager Thread Pool Max

Set the maximum number of execution threads that the engine can spawn simultaneously. Default is 300. A value of -1 means that there is no maximum number of threads.

This property does not appear if Process Server is configured to use an application server Work Manager.

If the number of threads being run is equal to this value, processes can fault as no threads are available when a node needs to broadcast information to other nodes. To be safe, you should create a secondary pool to be used by Process Server. This is done in the Application Integration Console. Process Server will only use threads in this pool when critical system work must be performed.

Work Manager Thread Pool Min

Set the minimum number of execution threads that the engine allocates for its work manager. Default is 25. Specify enough execution threads to run the number of processes plus the number of simultaneous invokes that processes might execute.

This property does not appear if Process Server is configured to use an application server work manager.

Work Manager Threads For Alarms Max

Set the maximum number of threads that the engine will use from the work manager to dispatch work scheduled by an alarm in a process. If there are 100's of alarms firing concurrently, all of the threads in the work manager could be used just to dispatch the alarms. If you experience performance issues or deadlocks because the alarm manager is using all of the threads, you can increase this value. Default is 5.

Work Manager Threads For Process Migration Max

Increase the thread count if you have hundreds of processes to migrate. This count is on top of execution threads for the server. Default is 50 and affects the number of threads allocated for new receives. For details, see the server property above, Migrate running processes even on warnings.

Work Manager Threads Per Process Max

Set the maximum number of execution threads the engine can spawn simultaneously for an individual process. Default is 10.

JMS Exception Listener

The Process Server establishes a JMS Exception Listener for each connection it maintains. If after the server established a connection, it receives a connection failure notification, it attempts to reconnect for a specified interval and number of attempts. Default is 30 seconds (30,000 milliseconds) for 20 attempts.)

Use the following SQL statements to change these values:

```
INSERT INTO AeConfigSettings (ConfigPath, ConfigValue)
VALUES ('MessagingManagers/<manager name>/ReconnectionInterval', '30000');
```

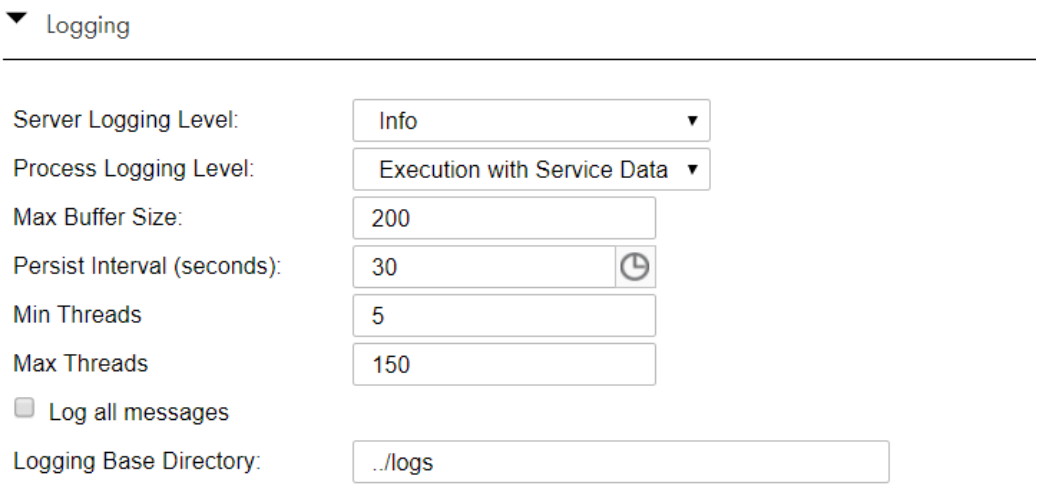
and

```
INSERT INTO AeConfigSettings (ConfigPath, ConfigValue)
VALUES ('MessagingManagers/<manager name>/ReconnectionAttempts', '20');
```

The only time you might want to change these values is when the default length of time is too short. If the Process Server cannot reconnect because the connection listeners are not notified, changing these settings won't help.

Logging Properties

On the Logging tab, you can select logging levels for the Process Server and for processes.



Property Name	Description
Server Logging Level	By default, Process Server generates a log of server events, including server property configuration changes, BPR deployments, server stop and start, and process failures. You can configure one of the following logging levels for the Process Server: - Info (default) - Error - Warning - Critical - Verbose (includes all levels) - Off (No logging occurs)
Process Logging Level	By default, Process Server generates an execution log for running processes. You can view or download an execution log for a running or completed process. An execution log provides start and end times for activity execution and helps you troubleshoot faulted processes. You can configure one of the following logging levels for running processes: - **None:** The Process Server does not log any information. Use this option to enhance engine performance. - **Execution:**This is the default option. The Process Server logs all execution statements except for *Will Not Execute* statements. Select **Execution** to decrease the size of the log file. - **Execution with Data:** The Process Server logs all execution statements except for *Will Not Execute* statements, but includes variable, expression, and partner link data. Select **Execution with Data** to decrease the size of the log file. - **Execution with Service Data:** The Process Server logs all execution and fault information, as well as some WSIO activity information. For execution information, Process Server logs deadpath states, terminations, ready-to-execute, and so on. For WSIO, Process Server logs invokes, picks, and receives, but excludes information related to data assignment and changes. - **Full**: The Process Server logs all execution statements *Will Not Execute* statements for deadpath activities. For example, the Process Server logs all fault handling statements that are not executed. To improve processing speed, perform no logging or minimum logging. Informatica recommends that you use the **None** or **Terse** logging levels.

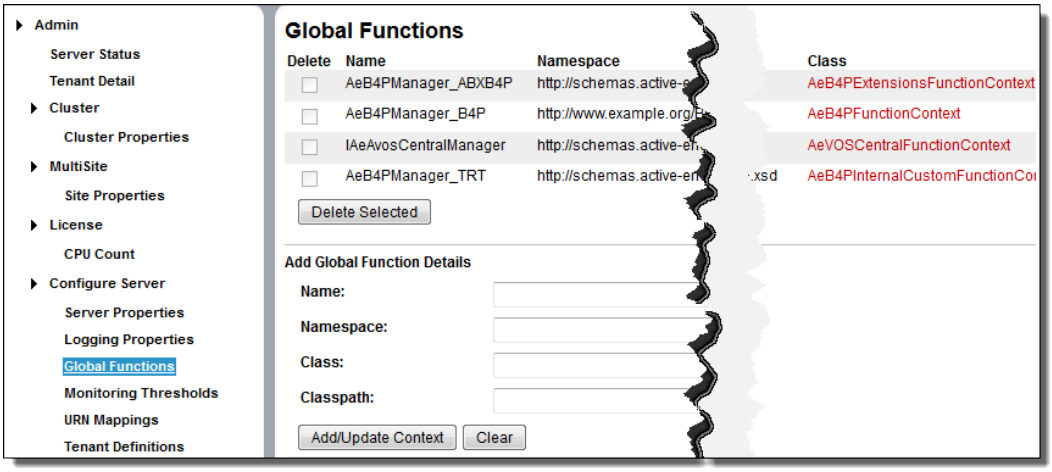
Property Name	Description
Max Buffer Size	The number of state changes, data changes, and other logging events that the Process Server holds in a buffer before writing them to the database. Default is 200.
Persist Interval (seconds)	The number of seconds that the Process Server waits before flushing the buffer and writing log events from memory to the database. The default value is 30.
Min Threads	The minimum number of threads that the Process Server initiates upon startup to log process events. Default is 5.
Max Threads	The maximum number of concurrent threads that can be active at any point of time for logging process events. Default is 150.
Log all messages	If enabled, logs data of each WS message going to and from the Process Server, including OData requests, and SOAP envelope, headers, and body. All HTTP messages are logged to the file system.
Logging Base Directory	Root directory for log files. The default base directory is the location pointed to by the `java.io.tmpdir` property. When messages are logged, each engine will create message log files under the root directory under `<BASE_LOGGING_DIRECTORY>/engine<id>/`. The configuration for the base logging directory can contain ant-style parameters that are resolved relative to system properties and environment variables. For example, `${CATALINA_HOME}/`logs will create log files relative to the `CATALINA_HOME` environment variable used to start your Tomcat server.

Global Function Contexts

On the Global Functions page, you can add custom function information for any process on the server. These functions are not within the scope of a contribution.

BPEL processes may contain custom functions that are used within XPath or other expression languages. Process Server provides a `FunctionContext` interface for implementation of custom functions. By using the `FunctionContext` interface, new or different functions may be installed and made available to the Process Server XPath (or another) expression writer.

If you already have custom functions implemented with a different interface, such as the jaxen `FunctionContext` interface, you can use them in your BPEL process.



If you have not yet written custom functions, you can download a BPEL process example from the Education Center on `http://www.active-endpoints.com`**. The Custom Function sample includes all the necessary source files for using custom function in Process Developer and on the Process Server.**

Compiling your Custom Function with Function Context Classpath References

To implement the Java-based `FunctionContext` interface, you'll need the files that are within:

```
[Process Developer product folder]\developer\plugins\org.activebpel.enginep_[version]
\server\shared\lib
```

You will use `org.activebpel.rt.bpel.function.IAeFunctionContext` contained within `ae_rtbpel.jar` in order to implement the `FunctionContext` interface. If you are implementing exception handling, you'll need `ae_rt.jar`.

Adding Global Functions to Process Server

You add custom function details to make the functions known to the engine.

You can specify an absolute classpath location for the function or use a system property to indicate a location that any server in a cluster can point to. You define the system property as appropriate for your application server. You'll add the following information to the Add Global Function Details section.

- Name—The name of the custom global function. This name appears in the Custom Function list. The name can be any identifier and need not match the name of the function that is actually used. Also there can be more than one function defined in the function context and have the implementation in the same jar. It is not necessary to create different entries addressing each of the functions.
- Namespace—Use the namespace identifying the function. The namespace is declared in BPEL processes using the function.
- Class—This is the container file that implements the custom function. This must be fully qualified.
- Classpath —The location for the custom function folder, zip or jar file. The custom function must be on the same machine as the server. The classpath can be an absolute path. For example:

  ```
  C:\apache-tomcat-6.0.20\activevos\lib\MyCustomFunction.jar
  ```

  In addition, it can be a system property. For example, a classpath using a JBoss system property value might look like:

  ```
  ${jboss.home}\MyCustomFunction.jar
  ```

  For dependent jars, add a semicolon delimited list, such as:

  ```
  ${jboss.home}\MyCustomFunction.jar;${jboss.home}\MyCustomFunction_dependency.jar
  ```

After adding this information, press the **Add/Update Context** button.

Process Server validates the function details and ensures that a class loader can load the class files.

If an error is reported, ensure that you have a valid class name and classpath location for the custom function on the same machine as the Process Server.

You can delete any function that you no longer need if you delete the associated processes.

Monitoring

Select engine properties to monitor on the Monitors tab. For each property, you can provide a statistic and threshold that, when reached, alerts you to a warning or error condition.

You can decide the frequency and interval of monitoring periods should be, as described in the following table.

Field	Explanation
Threshold Period	Period for collecting and aggregating statistics. The default is five minutes.
Evaluation Frequency	Number of times during the threshold interval the statistics are evaluated. For example, if the threshold interval is five minutes, and the evaluation frequency is five times, then statistics are collected and aggregated once a minute during every five-minute period. The default is 5 times.
Maximum Trouble Items	Number of error/warning items per engine to display on the Monitoring page of the Process Console
Monitor Alert Service	Described after this table.

Monitor Alert Service

Add the name of the service that will run when errors and warnings occur for monitored properties. This service also runs when it automatically triggered by a MultiSite site unavailable status. (*MultiSite* require a special license.)

When errors occur, Process Server instantiates the alert service, which can then invoke an action, such as notifying an administrator that a monitored property has an error condition. The service can also monitor engine status (running or stopped).

To add a service, type in the Service name, and select **Save**.

The service name is the My Role partner link service, identified in the PDD file deployed with the BPEL process to be used as the alert service. You can find this name by looking on the page.

After you add the service, select View Details to view the BPEL process.

Selecting Server Properties to Monitor

Use the following procedure to monitor a property, press the **Add Row** button.

Set the property by filling in this row, as follows:

- Property to Monitor: Select an item from the picklist.
- Level: Select a severity: Error, Warning, or Critical-Stop Server.
- Statistic: Select a statistic. Sometimes there is only one choice for a property.
- Op: Select a relational operator for the threshold.
- Threshold: Add a non-negative integer to be used in the evaluation.

After adding properties, select **Update.**

Monitoring Properties

You can monitor the following properties:

Cluster communications issue detected (count)

The number of times an issue with broadcasting cluster messages to nodes was detected.

Critical storage exceptions (count)

Storage exceptions include:

- Problems issuing select statements
- Network communications failure
- Database communications failure
- Database user permission problems

Database connection acquisition time (ms)

Tracks the amount of wait time to get a connection from the datasource. An excessively long wait may indicate signs of trouble with the size of the connection pool. The monitoring includes maximum and average values.

The Process Server storage layer does not perform connection pooling. It relies on the storage implementation (usually a `javax.sql.DataSource`) to pool connections. Consult your application server or database documentation to address any issues with poor performance related to the connection pool size or connection acquisition time.

Deadlock retry attempts (count)

A deadlock retry can occur when the engine attempts to:

- Lock a process
- Write process state or journal entries
- Remove alarms for dispatch
- Match inbound receives
- Acquire an internal counter value such as process Id or deployment Id

Discarded unmatched correlated receives (count)

When a message with correlation properties fails to route to a running process instance and is not able to create a new process instance, the engine will keep trying to dispatch the message for the configured amount of time. There is a limit to the number of such unmatched messages that the engine will retry. This property tracks the number of messages that were discarded due to the buffer of unmatched messages being full.

The unmatched correlated receive timeout, which is shown on the Server Properties page, controls the amount of time that the engine will keep the unmatched message queuing until it is routed to a process instance. The pool of unmatched receives may fill up if this timeout is too high. However, such a problem may be an issue with process design as opposed to the timeout or buffer size.

Engine removed from cluster (count)

The number of times an engine was removed from a cluster due to it going offline, missing cluster broadcast messages.

Failed to lock process (count)

In the event of a failover, a process instance that began on one cluster node may be moved to a different node during recovery. It is unusual, but possible, that a process lock may fail, causing a process to

suspend in the Suspended (programmatic) state. To address this problem, configure this property by set a Warning level for a count greater than 1 (or similar threshold).

By monitoring this property, as well as monitoring the Server Log for process recovery messages, you can determine if unwanted process suspensions are being caused by process lock failures.

Faulted/suspended(faulting) processes (count)

Number of processes that end in a faulted state or are suspended due to an uncaught fault.

You may have an expectation that processes running on this engine should not fault, and you want to be notified if they do.

Plan cache efficiency (percent)

Helps you determine if the Deployment cache setting is correct.

A deployment plan corresponds to each deployed version of a process, including associated disposition of running processes. Process versions that are active can be cached for better engine performance. The default number of plans that are cached is 100.

Plan cache removals (count)

The number of times process plans were removed from the cache.

Plan cache turnover (percent)

How often loading of new plans forced older plans out of cache.

Process cache efficiency (percent)

The engine configuration contains a count of the maximum number of processes which can be kept in memory before they are forced into storage.

A process is cached in memory if it is currently executing some logic or if it is quiescent but is being cached in anticipation of receiving another message, alarm, or response to an invoke. This property reports the percentage of processes that are read from memory versus process instances read from storage. For example, 100% indicates that all process reads are coming from the memory cache.

On the Server Properties page, you can set values for Process Count and Process Idle Timeout. The Process Count setting controls the size of the cache. The Process Idle Timeout setting controls how long to keep an idle process in the cache. If the process cache efficiency percentage is low and your processes contain bi-directional invokes or can process multiple inbound messages, you may benefit from increasing the Process Count and Process Idle Timeout. This will help keep processes in memory. However, if your processes are long-running and receive messages only periodically, a low process cache efficiency percentage is not necessarily a problem.

Process count exceeded (count)

The Server Properties include a Process Count option that specifies the maximum number of processes in memory. When the process count exceeds the value set for the Process Count, you can create an alert based on this property.

For example, if the Process Count is set to 50 and you want to be alerted when the count reaches 55, set this value to alert when the threshold is greater than or equal to 5.

Time to obtain plan (count)

The amount of time it takes to load a process plan into memory.

Time to obtain process (ms)

The time it takes to obtain a process is useful to determine if this operation is trending significantly higher under load situations. The monitoring includes maximum and average values. This property includes the time it takes to acquire a lock on a process as well as restore its state from storage if necessary.

This property works in conjunction with process cache efficiency.

Time to perform XSL transform (ms)

The time spent performing transforms within the `doXslTransform()` custom function.

Time to query Identity provider (ms)

Amount of time spent querying the identity provider in milliseconds. For example, if a request to the LDAP server to list groups for a user takes 20ms, that's the value of this metric for that instance.

This property allows you to track down slowness issues with an LDAP provider if you track a spike in time to query

Time to save process (ms)

Number of milliseconds required to save the process state and variables to the database.

A threshold can vary greatly depending on the process composition, number of variables, and size of variable data. This property only works for processes with a persistence setting of Full or Persist.

Time to validate messages (ms)

Reports the amount of time the engine spends validating input and output messages from receives, invokes and other activities.

This validation is enabled on the Configuration page labeled: "Validate Input/Output messages against schema." If enabled, all messages are validated. Validation can also be enabled or disabled on individual partner links through a policy assertion.

This property does not track the time spent in explicit variable validation through the BPEL validate activity or optional validate attribute on the assign activity.

You may wish to speed up processing by disabling all or selected message validation. If too much time is spent validating messages, you can take several steps. Start by redeploying processes with an added Message Validation policy assertion for partner links, which provides fine-grained control over specific types of messages. You can also disable message validation for all processes by disabling the Server Property.

Work manager work start delay (ms)

The time it takes between scheduling of a work item request and the actual start of work can help in tuning of the work manager pool.

If the time delay is trending upwards, there may not be enough threads available to handle the amount of work. The monitoring includes maximum and average values.

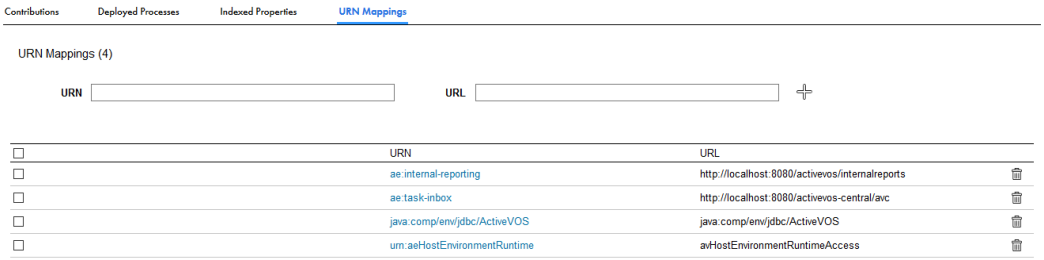
If you are using the default work manager, the size of the work pool can be configured on the Server Properties page. If you are using a work manager implementation provided by an application server, the size of the pool and the priorities of its threads should be configurable in your application server administration console.

URN Mappings

On the URN Mappings tab , there are system mappings and user-defined mappings.

You see four default mappings if you select a Secure Agent.

The following image shows a sample URN Mappings tab :



System Mappings

Process Server provides the following default URN Mappings for system services. You may need to manually add a mapping if it wasn't part of your initial configuration.

URN	URL
`ae:internal-reporting`	`http://localhost:8080/activevos/internalreports` Default address of the BIRT reporting engine for a deployed process containing a reporting service.
`ae:task-inbox`	`http://localhost:8080/activevos-central/avc` Default address of Process Central. Be sure to change the `host:port` to match your installed location, if needed.
`jjava:comp/env/jdbc/ActiveVOS`	This value is created using your specified JNDI database name
`urn:aeHostEnvironmentRuntime`	`avHostEnvironmentRuntimeAccess`

User-Defined Mappings

You can assign a physical address to a universal resource name (URN). The URN is a logical address of a partner link, specified in a deployment resource.

URN mappings provide a flexible and dynamic way to define target endpoint references. Use URN mappings to specify the physical address of a partner link endpoint reference instead of using the address specified in a process deployment descriptor (`.pdd`) file or WSDL file. By mapping a URN to a URL, you do not have to rely on invoking a statically defined endpoint address. URN mappings give you flexibility, for example, to deploy the same BPR files for testing and production environments.

Also, if you specify a URL, you can replace the URL by mapping it to a different URL.

The following example illustrates one type of URN to URL mapping:

```
urn:localhost = http://localhost:8080/active-bpel/services/${urn.3}
```

This mapping might be used when a process is deployed with the following partner link address information:

```
<partnerLink name="assessor">
   <partnerRole endpointReference="static"
     invokeHandler="default:Address">
      <wsa:EndpointReference xmlns:assessor="http://
       tempuri.org/services/loanassessor">
      <wsa:Address>urn:localhost:AssessRisk</wsa:Address>
      <wsa:ServiceName PortName=
        "SOAPPort">assessor:LoanAssessor</wsa:ServiceName>
   </partnerRole>
</partnerLink>
```

The Process Server invocation framework resolves the URN as follows:

```
urn:localhost:AssessRisk = http://localhost:8080/active-bpel/services/AssessRisk
```

Here are some ways you can map URNs to URLs. Note that each segment of the URN is separated by a colon.

URN	URL
`urnSegment1:urnSegment2`	`http://localhost:8080/active-bpel/services/MyService`
`http://ServerA:8080/active-bpel/services/MyService`	`http://ServerB:8081/active-bpel/services/MyService`
urn:localhost:service	`http://localhost:${AE-NODE1-PORT}/active-bpel/services/${urn.4}`

The last example in the table above shows how you can use variable substitution in an URL.

The URL values can optionally contain variables. The variables can be environment variables accessible through `java.lang.System.getProperties()` or a segment from the URN itself. The Apache Ant style variable declaration of `${property}` is used to identify a property within the URL. Segments from the input URN value can be referenced by using a special property naming convention of `${urn.offset}` where offset is a one-based offset identifying the segment from the input URN value to use for substitution.

The URL in the third mapping in this table contains two variables. The `${AE-NODE1-PORT}` variable pulls the port number from an environment variable. This variable would need to be set as a -D parameter on the Java runtime environment (for example, `java -D AE-NODE1-PORT =8080 ...`) or populated externally to the Process Server.

The `${urn.4}` variable, also in this example, references the fourth segment from the input URN value. Notice that the URN contains only three segments. The URN in the `.pdd` file should contain at least one other segment. A sample URN might be:

`urn:localhost:service:StoreService`.

As the value of the fourth segment of this URN is `StoreService`, the resulting URL is:

```
http://localhost:8080/active-bpel/services/StoreService/
```

Managing URN Mappings

Operations you can perform are:

- Add a new URN, fill in the URN and URL fields within the Add new URN Mapping section and then press the **Add/Update Mapping** button.
- Edit an existing URN, click on its link in the top area of this tab . Process Server fills in the URN and URL fields within the Add new URN Mapping section. After making changes, press the **Add/Update Mapping** button.
- To delete a mapping, select the check box next to the mapping and select **Delete**.

Identity Service

An identity service provides a way for a BPEL process to look up users and groups in an enterprise directory service or database. The Process Server identity service is based on one of the following:

- JDBC
- LDAP

- LDIF
- XML file

By providing the communication details for look-up access to your identity service, you can do the following:

- Run BPEL processes that implement identity-based activities. When a process runs, the person or group specified in the process is looked up in your identity service. Identity-based activities use a system-supplied WSDL, as described in the *Process Developer* documentation.
- For the BPEL for People functionality, specify users or groups to receive tasks into their Process Server or other task client

Note: When a user is removed from an ICS group and an executing process is associated with the group, there may be a short interval that the user can still invoke the process. This is because this information is cached and is refreshed every five minutes. This interval can be longer if ICS is not executing.

Creating a Trusted Keystore File

To ensure that the Process Server can trust the SSL certificate presented by the LDAP server, the LDAP server's certificate, or its CA's certificate such as VeriSign, must be installed in a Java key store file which is designated as the store keeping a list of trusted certificates. This trust key store file must be accessible by the Process Server engine.

Example

Assuming your LDAP server is Apache DS running SSL using a self-signed certificate, you would use the following steps to create a trusted keystore file.

1. Export the Apache DS SSL certificate as a DER-formatted file using the Java/Sun keytool. For example:

   ```
   c:> keytool -export -keystore apacheds.ks -alias apacheDsAlias
              -file aeldap.cer
   ```

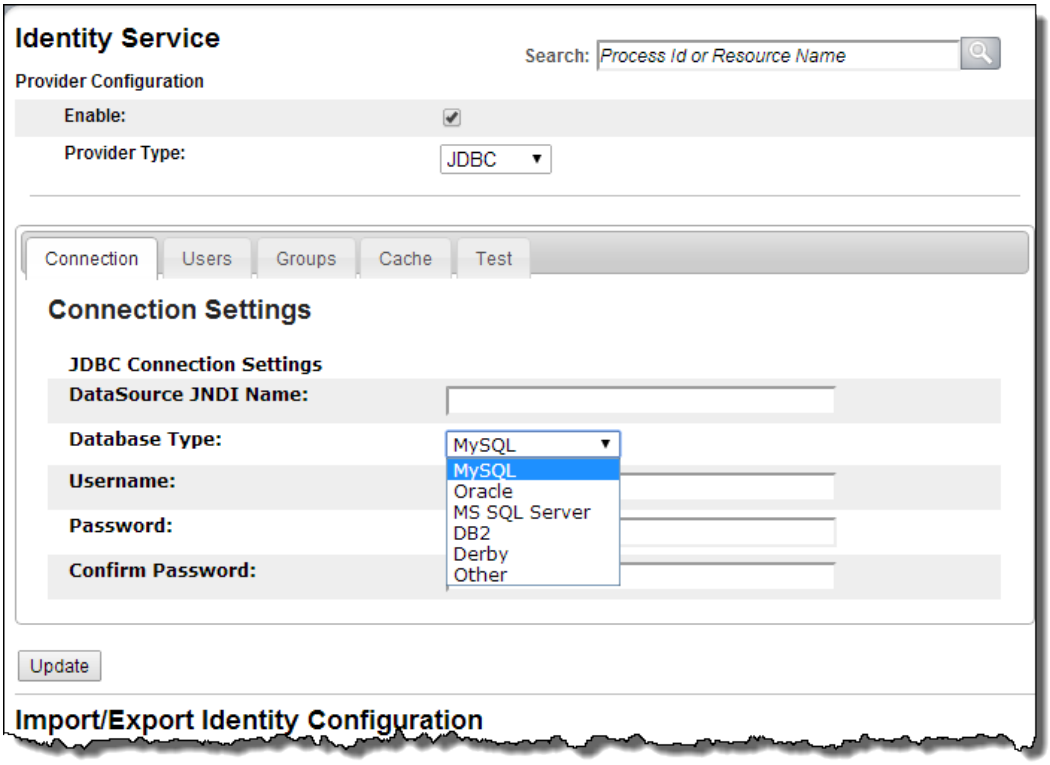
   where `apacheds.ks` is the key store database in which the Apache DS SSL certificate is stored, *apacheDsAlias* is the alias within that key store, and *aeldap.cer* is the name of the file where the certificate is exported to.
2. When you run the command in Step 1, you will be prompted for the keystore password. Add the password for the `apacheds.ks` file.
3. Create a new keystore file, such as `aeTrustedCA.ks`, on the same machine as the Process Server engine. This key store will contain trusted certificates. Also in the same command, import the certificate. For example:

   ```
   c:> keytool -import -file aeldap.cer -alias apacheDsAlias
              -keystore aeTrustedCA.ks -storepass secret
   ```

   where `aeldap.cer` is the file that was exported from the Apache DS server `aeTrustedCA.ks` is the name of the new key store file on machine that is running the Process Server engine, and *secret* is the password for the `apacheds.ks` file.
4. When you run the command in Step 3, you will be asked whether to trust this certificate. Type in *yes* to add the certificate.
5. From the Identity Service Page of the Process Console, enter the full path to the `aeTrustedCA.ks` file for the Trusted Keystore Path.

JDBC Identity Service

If you select a Provider Type of JDBC, the Connection tab is as follows:



In your identity service, be sure to map the Process Central security role, `abTaskClient`, to each user that will login to Process Central.

View and update identity service settings as shown in the table below.

Before enabling a JDBC datasource for the identity service ensure the prerequisites are met:

- A database is configured for use with JDBC identity service
- A connection pool resource is configured in your application server
- The database driver exists in the `common/lib` (or similar) directory

Note: Tomcat users must manually configure the JNDI `DataSource` in the Process Server context file.

If desired you can import and export your JDBC configuration.

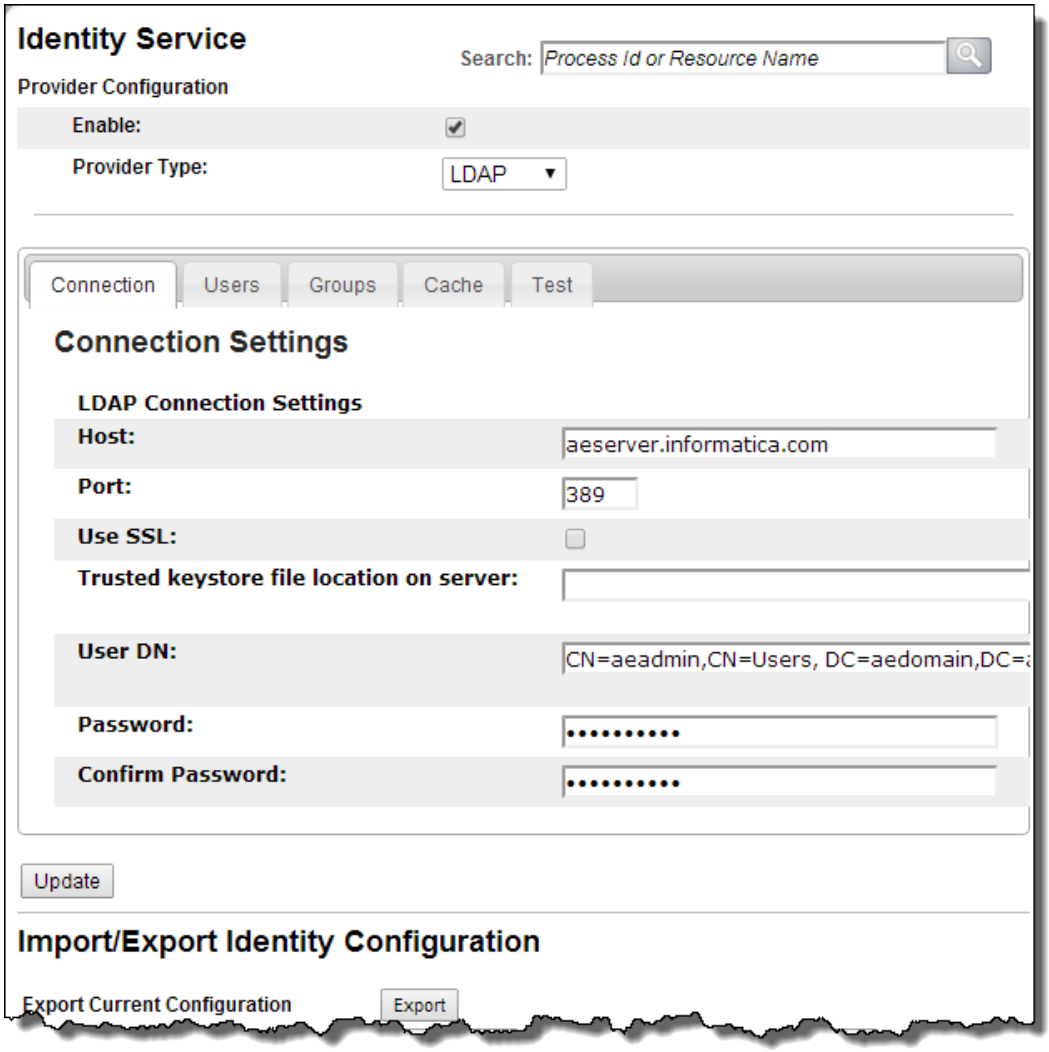
Enter the provider-specific connection settings used to establish connectivity to your identity store.

Provider Configuration	
Enable	Add a checkmark to use the identity service. Initially the service is disabled since it is not configured and ready for use. Configure the remaining settings, and then enable the service.
Provider Type	Select JDBC from the drop-down list:
DataSource JNDI Name	The resource must be assigned a JNDI name for location at run-time. For example, `java:comp/env/jdbc/IdentityTest`
Database Type	Choose database type from picklist. Choose Other to use generic settings

Provider Configuration	
Username	(Optional). An application server may require credentials for a JNDI context; that is, credentials for doing a lookup on the JNDI tree. Your application server may require credentials for looking up a connection pool for access to the identity service. In this case, you must provide the user name and password. For a local data source, you can leave this field blank. Note that the user name for the JNDI context is not the same as the user name for the data source itself. The data source username must be set on the connection pool resource.
Password	(Optional) Password associated with Username

LDAP Identity Service

If you are using an identity service that makes use of LDAP, select it from the Provider Type picklist. Here is what you will see:



Note: In your identity service, be sure to map the Process Central security role, `abTaskClient`, to each user that will login to Process Central.

View and update identity service settings as shown in the following table. If desired, you can import and export your LDAP configuration.

Enter the provider-specific connection settings used to establish connectivity to your identity store.

Provider Configuration	
Enable	Add a checkmark to use the identity service. Initially the service is disabled since it is not configured and ready for use. Configure the remaining settings, enable the service, and select **Update**.
Provider Type	Select LDAP from the drop-down list:
Host	Enter the LDAP server's DNS name such as `ldap1.my-domain-name.com` or IP address such as 192.168.1.1.
Port	Enter the port to use for communications between the Process Server and the LDAP server. The default value is 389.
Use SSL	(Optional) Enable this checkbox to provide encrypted transport communication between Process Server and the LDAP service. If you enable this, you must enter a trusted keystore file location in the next field.
Trusted keystore file location on the server	(Optional) Enter the full path to the a`eTrustedCA.ks` file for the Trusted Keystore Path. This file must be accessible by all instances of the server when deployed in a clustered environment. This path is required if SSL is enabled. This is discussed following this table.
User DN	Enter the user distinguished name. Most directory servers do not allow anonymous access, therefore the username and password is required. The username should be the distinguished name of the user. For Microsoft Active Directory, an example of a user distinguished name is: `CN=Administrator, CN=Users, DC=domainname, DC=com` (or local) For an open LDAP service, an example of the DN is: `uid=admin, ou=system`
Password	Enter the administrator password for access to the directory service, and confirm it.

Creating a Trusted Keystore File

To ensure that the Process Server can trust the SSL certificate presented by the LDAP server, the LDAP server's certificate, or its CA's certificate such as VeriSign, must be installed in a Java key store file which is designated as the store keeping a list of trusted certificates. This trust key store file must be accessible by the Process Server engine.

Example

Assuming your LDAP server is Apache DS running SSL using a self-signed certificate, here's how you could create a trusted keystore file.

1. Export the Apache DS SSL certificate as a DER-formatted file using the Java/Sun `keytool`. For example:

   ```
   c:> keytool -export -keystore apacheds.ks -alias apacheDsAlias
             -file aeldap.cer
   ```

   where `apacheds.ks` is the key store database in which the Apache DS SSL certificate is stored, `apacheDsAlias` is the alias within that key store, and `aeldap.cer` is the name of the file where the certificate is exported to.

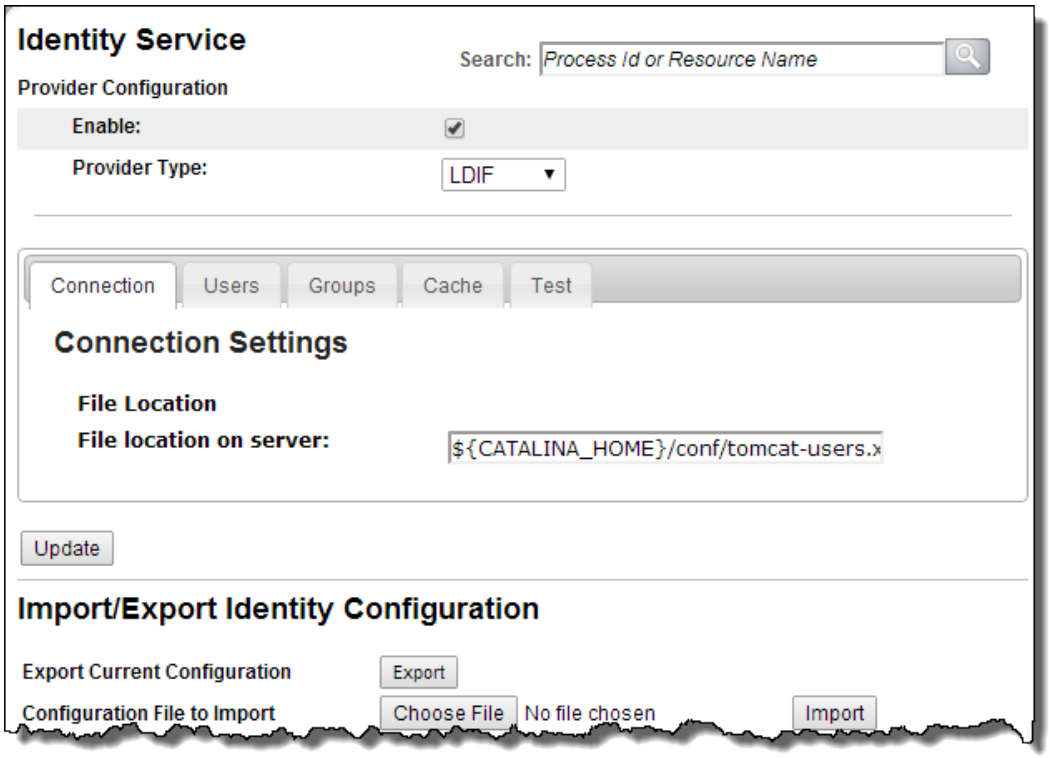
2. When you run the command in Step 1, you will be prompted for the keystore password. Add the password for the `apacheds.ks` file.
3. Create a new keystore file, such as `aeTrustedCA.ks`, on the same machine as the Process Server engine. This key store will contain trusted certificates. Also in the same command, import the certificate. For example:

```
c:> keytool -import -file aeldap.cer -alias apacheDsAlias
          -keystore aeTrustedCA.ks -storepass secret
```

   where `aeldap.cer` is the file that was exported from the Apache DS server `aeTrustedCA.ks` is the name of the new key store file on machine that is running the Process Server engine, and *secret* is the password for the `apacheds.ks` file.
4. When you run the command in Step 3, you will be asked whether to trust this certificate. Type *yes* to add the certificate.
5. From the Identity Service Page of the Process Console, enter the full path to the `aeTrustedCA.ks` file for the Trusted Keystore Path, as described rarlier in this topic.

XML or LDIF Identity Service

If you are using an identity service that makes use of XML or LDIF, select one of these from the Provider Type picklist. Here is what you will see:



In your identity service, be sure to map the Process Central security role, `abTaskClient`, to each user that will login to Process Central.

View and update identity service settings as shown in the following table.

Note that file-based identity service does not require any additional information beyond the connection settings.

Enter the provider-specific connection settings used to establish connectivity to your identity store.

Provider Configuration	
Enable	Add a checkmark to use the identity service. Initially the service is disabled since it is not configured and ready for use. Configure the remaining settings, enable the service, and select **Update**.
Provider Type	Select from the drop-down list: - **LDIF**. (LDAP Data Interchange Format). A file-based service representing (LDAP) directory content. - **XML File**. A file-based service, such as the entries found in `tomcat-users.xml` file. Be sure to see the notes that follow this table.
File location on server	This field is displayed only if the Provider Type is LDIF or XML File. Do one of the following: - For an LDIF provider, provide a path to your `.LDIF` file. (See the notes that follow this table.) - For a provider based on XML user entries, provide a path to your file. (See the notes that follow this table.)

Notes for File-Based Service:

- If you are running in a clustered environment, ensure that the file is shared or is on the same path in all servers.
- The file you select must be on the Process Server
- You can add an email entry to your XML file. For example, `<user name="user1" password="user1pw" roles="admin,oabTaskClient" email=”user1@example.com”/>`
- The LDIF or XML file is read each time the Process Server starts. You can update the file at any time. If the server is running, select **Update** to refresh the file contents.

User and Group Attribute Mapping

When looking up a user or group in an LDAP or JDBC-based Identity service, Process Server search procedure includes several basic identity attributes. For users and groups, these are:

- **Users**
  - person
  - memberOf (recommended, if Identity service supports it)
  - userName (required)
  - email
  - firstName
  - lastName
- **Groups**
  - groupName (required)
  - member (required for LDAP)

This generic model applies to any Identity service, and you can use it as is, or delete the optional attributes from the model.

The `memberOf` Users attribute is recommended for making searches for group members more efficient, especially where a user is a member of more than one group. Be sure to map a user as `memberOf` all relevant groups and add the user as a `member` in relevant Groups.

If desired, you can add many other search attributes to the basic model, and then use these attributes in LDAP or JDBC people queries from within a BPEL process People activity. When you add a search attribute from your Identity service, you must map it to a new attribute that gets added to the Process Server search model.

For example, if your Identity service includes a `loginCount` attribute, you can add that to the Users or Groups attribute mapping page. The attributes can be loaded into the Process Developer Process Deployment Descriptor Editor, where a developer has access to them for creating Logical People Group queries. For details on using identity attributes in Process Developer, see *Logical People Group Queries* in *Process Developer* Help.

To Add an Attribute Mapping:

1. On the Users or Groups tab of the Identity Service, select **Add Attribute**.
2. In the Provider Attribute/Column Name column, type in the name of an existing attribute that is in your identity store.
3. In the Model Attribute column, type in the same name or alias for the attribute.
4. Select the data type from the list. The list contains all types defined by the search model, based on the Higgins Open Source Identity Framework. (The Process Server uses Higgins to enable the adding of identity attributes to the search model.) Note that the list also contains two custom types, `GroupReference` and `PersonReference`. Use one of these types if you want to reference a group or user by name, rather than by the full distinguished name or primary key defined in the data store.

To Delete an Attribute Mapping:

1. Select the checkbox next to the Model Attribute field.
2. Select **Update**. The attribute is removed.

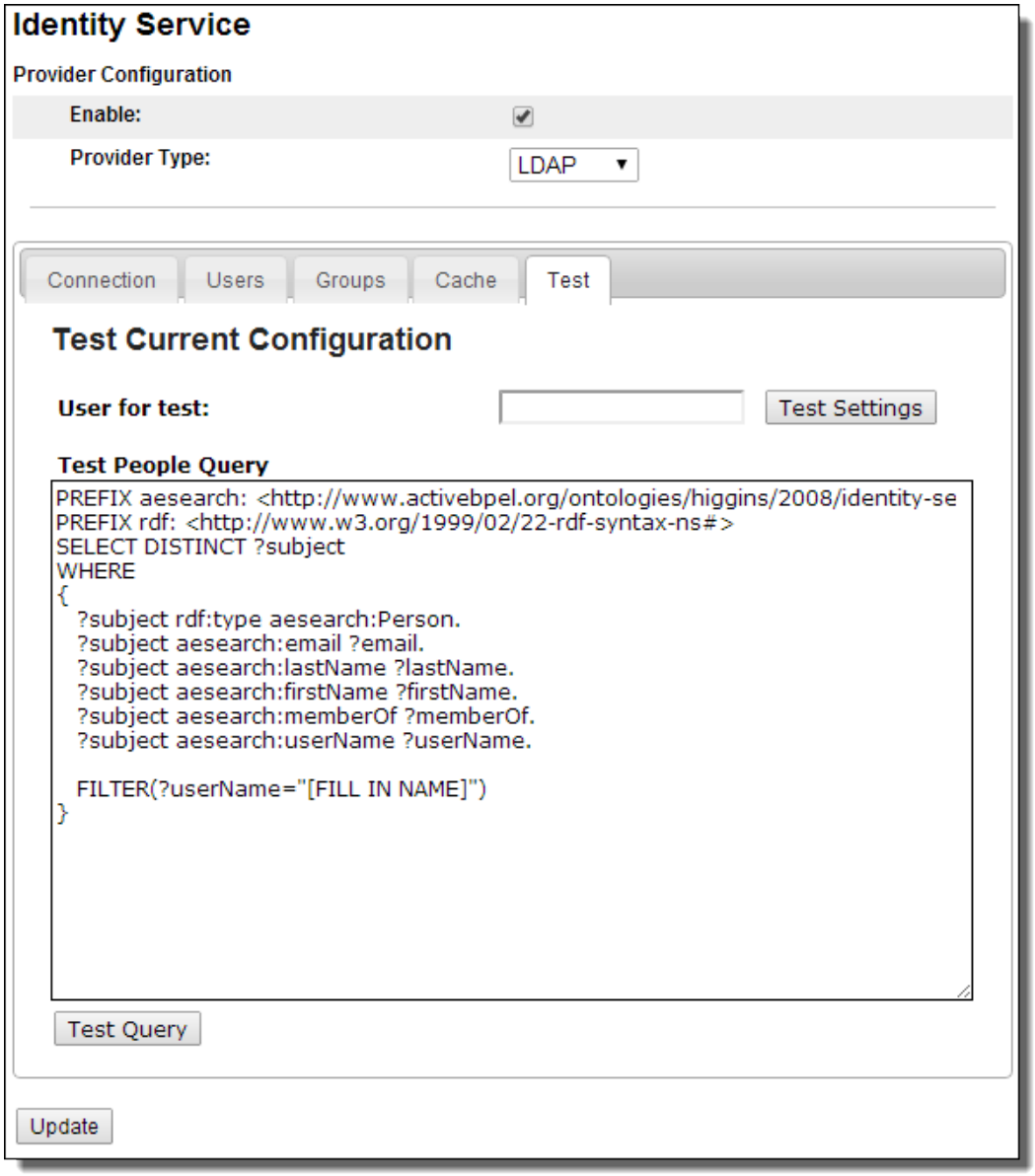
Testing LDAP and JDBC Configurations

You can test user and group configurations for an Identity service in the following ways:

- Select a user for testing
- Write a test query

Note: The Test tab is the same for JDBC and LDAP.

After selecting this tab, the following is displayed:



Selecting a user for testing

In the Test tab, enter a user name listed in your Identity service, and select **Test Settings**. This test ensures that the Identity service is set up correctly. If the test fails, the error messages try to suggest how to correct the configuration settings.

Writing a test query

The Process Server Identity search model allows identity queries to be made using the SPARQL Protocol and RDF Query Language. SPARQL is a standard language, well-suited to query identity resources. The Process Server use of SPARQL includes preconfiguring most of the required parts of the query. You need only to add a FILTER keyword to test the user and group attributes you added on the Users and Groups tabs of the Identity Service configuration page.

By writing the FILTER clause of a SPARQL expression, you can test the results of required and optional attributes. The default FILTER clause in the Test Query box is: `FILTER($userName="[FILL IN NAME]")`. Replace the text between the quotes with a user name and select Test Query.

For details on writing SPARQL queries, see `http://www.w3.org/TR/rdf-sparql-query/`

Sample SPARQL Queries

- `FILTER (($memberOf = "NE_Reps") || ($memberOf="NW_Reps"))`

  Select users that are members of either the "NE Reps" or "NW Reps" group
- `FILTER ( (($memberOf = "NE_Reps") || ($memberOf="NW_Reps")) && BOUND(?email) )`

  Select users that are members of either the "NE Reps" or "NW Reps" group and have an email address
- `FILTER($vacation != "On Vacation")`

  Select users who are not on vacation.

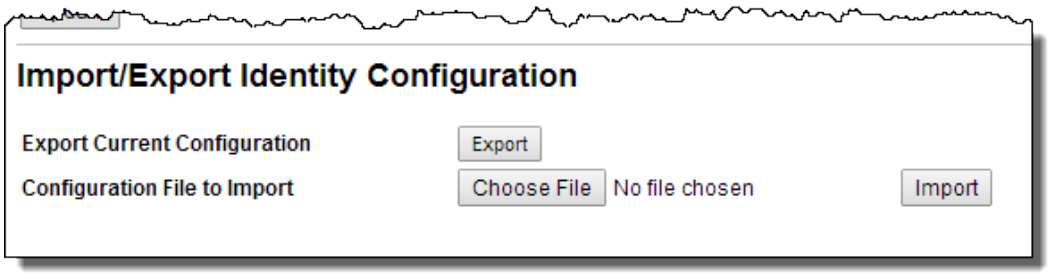
Note that the Test Query box accepts only literals for testing.

See also *User and Group Attribute Mapping*.

If your JDBC Identity service is on Apache Tomcat, see *Manual Configuration for a JDBC Identity Service on Tomcat*.

Importing and Exporting an Identity Configuration

The Import/Export Identity Configuration section is at the bottom of all Identity Service pages.



On the Connection Settings tab of the Identity Service, select **Export** to generate an XML document displayed in your browser. You can use the browser's Save As feature to save the document to file. You can then import these settings to another Process Server.

Note: On Google Chrome, you'll see a blank page. Right-mouse click and select *view page source* to view the XML document.

Import a configuration file by browsing to the file and selecting Import.

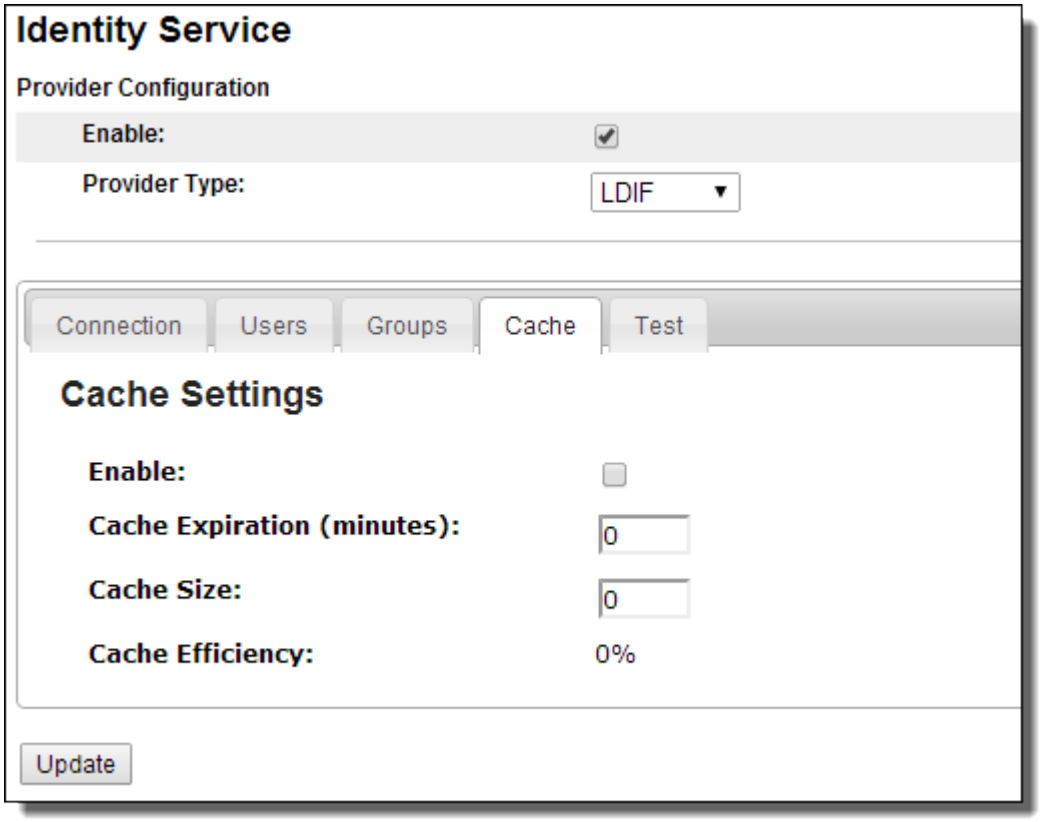
Identity Service Cache Settings

To enhance server performance, Process Server caches query results from the identity service for running processes. By default, the cache is enabled.

Note: The same information is displayed for all identity services.

If you do not expect many changes to the identity service as you run processes, you should enable caching. If the members of your identity service change frequently, you can increase the Cache Expiration and Cache Size values or disable the cache.

After selecting the Cache tab, the following is displayed:



1. Select **Enable** to use cache settings.
2. Set the Cache Expiration to at least one minute. The maximum is unlimited and the default is 30 minutes.
3. Set the Cache Size to a unit of in-memory queries to the identity service. The default is 100.
4. Press **Update**.

After enabling or disabling the cache settings and selecting **Update**, Process Server immediately makes the change.

This page also displays the cache efficiency. This is the percentage of reads from in-memory cache compared to total reads. Total reads includes going out to the identity service get the data. A large percentage value indicates that the in-memory cache is being used.

Manual Configuration for a JDBC Identity Service on Tomcat

To use a JDBC Identity Service on Tomcat, you must perform a manual step in addition to configuring the service in the Process Console. You must configure the JNDI data source by adding a declaration for your resource to the Process Server context.

Add an element like the following example to two files: `$CATALINA_HOME/conf/Catalina/localhost/activevos.xml file` and `active-bpel.xml`.

Nest the declaration inside the Context element.

Example:

```
<Resource name="jdbc/TestDB"
   auth="Container"
   type="javax.sql.DataSource"
   maxActive="100"
```

```
maxIdle="30"
maxWait="10000"
username="javauser"
password="javapw"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/javatest?autoReconnect=true"/>
```

JMS Messaging Service

Create a JMS Messaging Service to use a JMS provider and connect to a JMS queue or topic.

JMS Messaging Service Properties

Configure the following properties for a JMS Messaging Service:

Property	Description
Default JMS Provider	Indicates whether the selected JMS provider must be used as the default JMS provider. When you create multiple JMS Messaging Services, you can designate one JMS provider as the default for all the services.
JMS Provider Type	The JMS provider type that is to be used for the JMS Messaging Service. You can select one of the following JMS providers: - JBoss Messaging - JBoss MQ - Oracle Weblogic - IBM MQ Series - IBM WebSphere - Other JMS - Other JCA When you select a JMS provider type, a list of initial context properties that contain pre-defined default settings for the selected JMS provider appears. You can update the values that appear in the **Initial Context Properties** section. You can specify any provider that provides JNDI access to JMS resources. If the provider that you want to use is not available in the list, you can select **Other JMS** to populate the configuration with commonly used, generic JNDI properties. If you select the **IBM WebSphere** provider type, you will not be able to configure the queue listener and topic listener properties and the following message appears: `Configuration of Queue and Topic listeners is not available with the Websphere JMS provider. Websphere JMS explicitly forbids creating asynchronous consumers outside of a message-driven bean deployment when running on a Websphere server.`
Connection Factory Name	The JNDI name of the JMS connection factory.
Connection User	Indicates whether to include user credentials while creating connections on the connection factory. If the provider is hosted by an application server, you do not need to configure the connection user because authentication takes place when you access objects through the JNDI context. IBM MQ Series is a provider that requires connection credentials.

Property	Description
Connection Password	Password for the connection user.
Send Empty Credentials	This option is applicable for IBM MQ Series when you connect to a remote queue manager and do not enable authentication. You must send empty strings ("") in the connection for the user name and password. Otherwise, the connection fails with a security exception.
Maximum Total Connections	Each JMS Manager maintains an internal pool of connections to enhance performance when interacting with a remote JMS provider. This option sets the maximum number of active connections allowed at any time. This includes connections used for asynchronous listeners configured under **Queues and Listeners**, **Topics and Listeners**, and those used for invoke activities. If the maximum number is reached, clients must wait until a connection is returned to the pool. Setting the maximum value to -1 indicates that the manager may create as many connections as needed, with no upper limit so clients never have to wait. Change this value from the default of -1 if more connections are being created than the JMS provider can handle. Configure one of the following values: - -1 (unlimited) - A number higher than the total number of listeners for queues, topics, and some connections for invoke activities Default is **-1**.
Maximum Free Connections	Sets the number of unused connections the manager retains at any one time. If a connection is returned to the pool, and there are already the maximum free connections sitting idle, the connection is closed. This allows the total number of connections from the pool to shrink and grow as necessary. Setting the maximum free connections to 0 prevents the Process Server from holding onto connections. Each client receives a newly created connection. Ensure that the value of the **Maximum Free Connections** field is lower than the value of the **Maximum Total Connections** field. Default is **15**.
Delivery Mode	This option controls whether or not the JMS provider persists messages to storage for all processes. An individual process can have a different persistence setting, which overrides this option. Enable this option to persist messages in the event of a JMS failure. When this mode is enabled, Process Server instructs the JMS provider to ensure that a message is not lost in transit in case of a JMS provider failure. It is logged to stable storage. Note that persistent delivery requires that your JMS provider be configured with storage. Also, there is usually a performance hit with persisting messages. Default is **Persistent**.
Time To Live (ms)	Specifies the amount of time an unconsumed message remains on a queue. If a message will become obsolete after a certain period, you might want to set an expiration time. The expiration of obsolete messages conserves storage and computing resources. Default is **0 milliseconds**, which means that the default for the provider is used. This means that messages never expire and remain on the queue forever.
Priority (int)	Specifies a non-negative integer for a message handling priority. Default is **0**, which means that the default for the provider is used. An individual process can specify a priority, which overrides this option. JMS defines a 10-level priority value with 0 as the lowest and 9 as the highest. Clients must consider 0-4 as gradients of normal priority and 5-9 as gradients of expedited priority. Priority is set to 4, by default.

Property	Description
Maximum Reconnect Attempts	Specifies the maximum number of reconnection attempts that the JMS Manager must make when it tries to connect to the JMS provider. Enter **-1** for unlimited reconnection attempts. Default is **20**. When you stop the JMS Manager, the reconnection attempt is also stopped.
Reconnect Interval (ms)	Specifies the number of milliseconds to wait between each connection retry attempt. For example, to retry to connect up to 10 times with a 10000 millisecond delay between retries, set the **Maximum Reconnect Attempts** field to **10** and the **Reconnect Interval (ms)** field to **10000**. Default is **30000 milliseconds**.

Queue Listener and Topic Listener Properties

Configure the following properties for queue listeners and topic listeners:

Property	Description
JNDI Location	Location name used for JNDI look-ups.
Listener Class	Message listener class name. The listener class dispatches messages to Process Server. To deploy processes created with Process Developer, use the listener class name as **com.activee.rt.mom.jms.transport.AeJmsBpelListener**. If needed, you can create a custom listener class that extends the default listener class to include custom behavior. Specify the custom listener in this field. Default is **com.activee.rt.mom.jms.transport.AeJmsBpelListener**.
Listener Count	Number of connections to keep open. When the server starts, the JMS manager creates instances of this class that serve as asynchronous consumers on the destination. Each asynchronous consumer has its own connection to the JMS server. The number of consumers and connections created is controlled by the listener count. Default is **15**.
Selector	JMS message selector string.
Default Service	Specifies the name of the BPEL service to use when the target service cannot be determined from the addressing headers or message properties. **Note:** One alternative to specifying a service is to include the following JMS message property in the incoming message: `JmsTargetService = "myRoleServiceName"` Another alternative is to include a service name as a query parameter in a wsa: To header in a myRole partner link (for example, `<destination JNDI name>?<servicename>`). These scenarios are for receiving XML messages over JMS.
Run-As Identity	Specifies an identity for a JMS request. The identity is a role-based or group membership. If not specified, the request runs anonymously. Using this property is analogous to specifying an identity for a message-driven bean in `ejb-jar.xml`. Use this option if you want the JMS listener to invoke processes that have been secured with the **Allowed Roles** restriction in the Process Deployment Descriptor (pdd). Another use case is using the aeRunAs header for Human Task (B4P) operations. This case allows you to specify a RunAs Identity with the abTrust security role.

Property	Description
Tenant Context	Applicable for a MultiTenant-licensed server.
XA Transaction	If enabled for a supported platform, and if the required setup has been performed, indicates that a JMS read from the queue and the process engine must participate in the same transaction.

Rollback On Error	Select the **Rollback On Error** option if you want to decline messages that encounter an exception, or failed messages, and move to the next message. For example, the exception could be because of an authentication failure.
	If you select this option, you must configure how the message server must handle failed messages. For example, you can configure the message server to redeliver failed messages for a specified number of times. Or, you can configure the server to move failed messages to a dead letter queue and take action later. For example, add the following policy entry to configure Apache ActiveMQ server to move failed messages to a dead letter queue:
	`<policyEntry queue=">">`
	`<deadLetterStrategy>`
	`<individualDeadLetterStrategy queuePrefix="DLQ." useQueueForQueueMessages="true" processNonPersistent="true"/>`
	`</deadLetterStrategy>`
	`</policyEntry>`
	Here, if the original queue is `userIDs`, the dead letter queue that Apache ActiveMQ server creates is `DLQ.userIDs`.
	The **Rollback on Error** option does not apply to messages that were delivered to target processes. A message is considered successfully processed if it invokes a target process, even if the process fails.
	For new queues and listeners, the **Rollback on Error** option is selected by default.

Creating a JMS Manager

To create a JMS Manager, specify properties for the JMS Messaging Service and the queues or topics that the service must listen to.

Before you create a JMS Manager, you must copy the required client `.jar` files to the location specified in the classpath and restart the server.

1. Click the **ADMIN** tab.
2. On the left navigation pane, expand the **Configure Services** section and click **Messaging Service**.

   The **Messaging Service** page appears on the right pane.
3. Enter a name in the **Manager Name** text box and click **Add Manager**.

   The JMS Manager appears in the Manager list. The status is set to **Paused**.
4. Click the JMS Manager name.

   A page appears where you can configure properties for the JMS Messaging Service, queue listeners, and topic listeners.
5. Configure the properties for the JMS Messaging Service in the **JMS Messaging Configuration** section.

   For information about JMS Messaging Service properties, see "JMS Messaging Service Properties" on page 79.
6. To specify additional initial context properties, perform the following steps:
   a. Click **Add Property** under the **Initial Context Properties** section.

b. Specify a name and value for the property.

You can specify multiple initial context properties. You can also update the existing initial context properties for the selected JMS provider.

7. To add a queue listener, perform the following steps:
   a. Click **Add Queue** under the **Queues & Listeners** section.
   b. Specify a name for the queue listener.
   c. Configure the queue listener properties.

      For information about queue listener properties, see "Queue Listener and Topic Listener Properties " on page 81.

   You can specify multiple queue listeners.
8. To add a topic listener, perform the following steps:
   a. Click **Add Topic** under the **Topics & Listeners** section.
   b. Specify a name for the topic listener.
   c. Configure the topic listener properties.

      For information about topic listener properties, see "Queue Listener and Topic Listener Properties " on page 81.

   You can specify multiple topic listeners.
9. Click **Update** to create the JMS Manager with the specified properties.

After you create the JMS Manager, you can start it.

Updating a JMS Manager

After you create a JMS Manager, you can update its properties if needed.

1. Click the **ADMIN** tab.
2. On the left navigation pane, expand the **Configure Services** section and click **Messaging Service**.

   The **Messaging Service** page appears on the right pane displaying a list of JMS Managers.
3. Click a JMS Manager from the list.
4. Edit the required properties.

   To delete a queue listener, topic listener, or initial context property, select the corresponding check box.
5. Click **Update** to update the JMS Manager properties.

Starting, Stopping, Restarting, and Deleting a JMS Manager

After you create a JMS Manager, you can start, stop, restart, and delete it if needed.

1. Click the **ADMIN** tab.
2. On the left navigation pane, expand the **Configure Services** section and click **Messaging Service**.

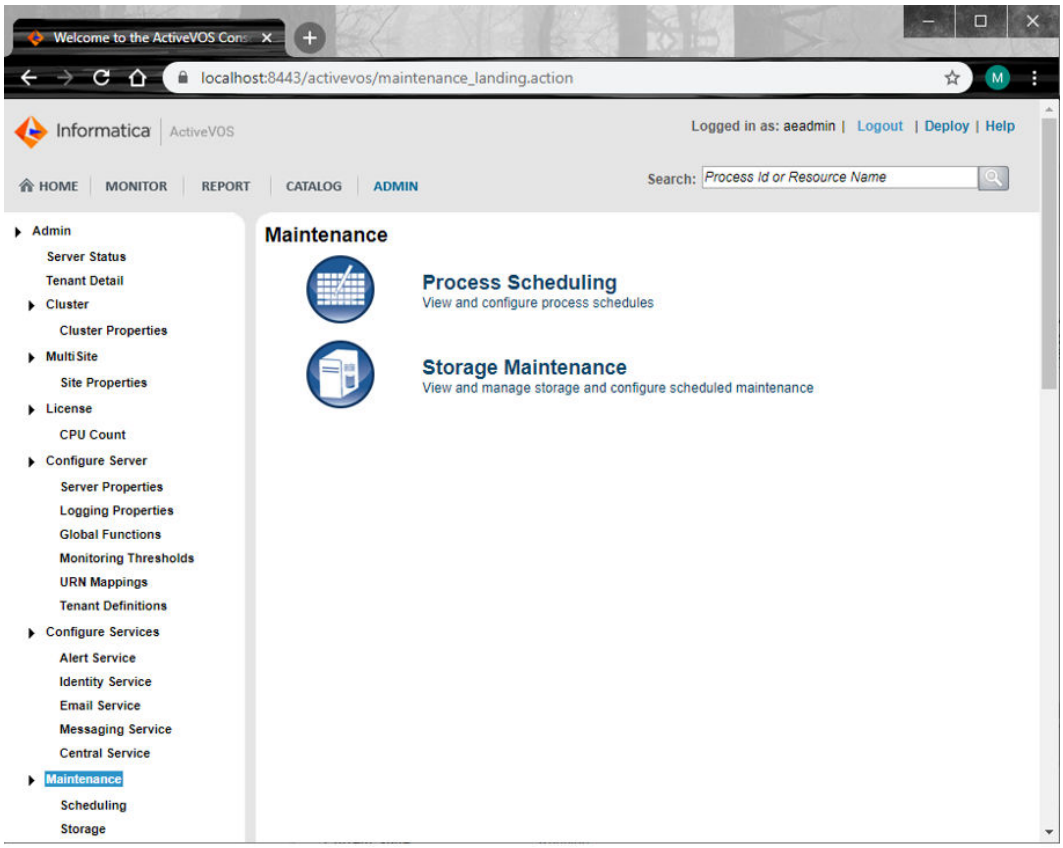
   The **Messaging Service** page appears on the right pane displaying a list of JMS Managers.
3. Select a JMS Manager from the list.

4. Perform one of the following actions:

Option	Description
To start a JMS Manager	Select **Start** from the **Actions** list, and then click **Execute**.
To stop a JMS Manager	Select **Stop** from the **Actions** list, and then click **Execute**.
To restart a JMS Manager	Select **Restart** from the **Actions** list, and then click **Execute**.
To delete a JMS Manager	Select **Delete** from the **Actions** list, and then click **Execute**.

Maintenance

After selecting this command, the following page is displayed:



Maintenance topics are as follows:

- Scheduled Processes
- Creating a New Process Schedule
- Managing and Modifying Execution Schedules
- Storage

Scheduled Processes

A BPEL process may be created to perform routine maintenance or other recurring task. If desired, you can deploy the process and then set a schedule to run it, such as weekly or on a specific day of the month. You can set a fine-grained schedule (intraday, seconds), if desired.

View the Scheduled Processes page from the **Admin > Maintenance > Scheduling** menu.

The following table describes information you'll see on this page.

Field	Explanation
Enabled	A checkmark next to a schedule name indicates the process is enabled for execution
Name	Descriptive name of the schedule. For example, use the process name, with or without other descriptive details
Service Name	A service from the **Service Definitions** list. Select the Service Name to view the service definition. Hover help displays the process name associated with the service.
Last Status	Execution state of the active process, such as Completed or Running. Select the Status to view the Active Process Details page. Hover help displays the Process Id.
Last Execution	Date and time of the last process execution, if any
Next Execution	Date and time of the next scheduled execution, if the schedule is enabled

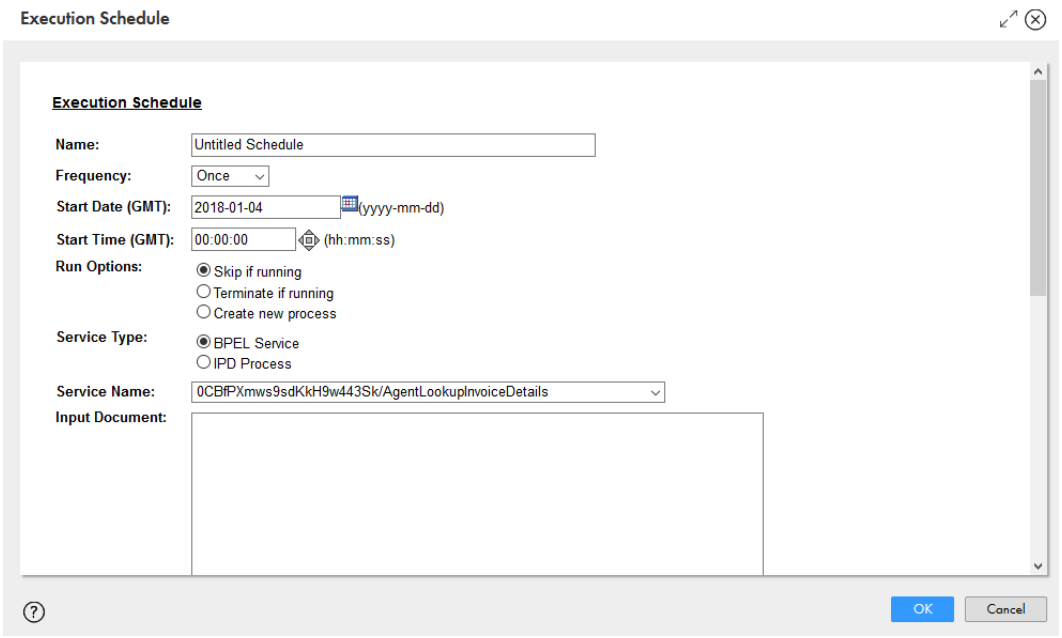
In a clustered environment, one of the nodes acts as an scheduling master at any time. Scheduled processes are elected to run only on the schedule master node. The schedule node is elected based on the server that starts up first. Only if there is a drop in the cluster communications do other nodes choose to get elected as the schedule master. Also Process Server does not maintain a history of the execution of scheduled processes.

To create a scheduled process, click the **Create Schedule** link at the top of the page. Process Server displays a dialog box that is described in *Creating a New Process Schedule*.

See also *Managing and Modifying Execution Schedules*.

Creating a New Process Schedule

To create a scheduled process, click **Add** at the top of the **Process Schedule** page. You see the following dialog box:



Execution Schedule	
Name	Provide a name to appear in the schedule list.
Frequency	- **Once**. Select this to execute on the start date and time specified. **Tip:** This option is good for running a test execution. You can then duplicate the details for a different execution schedule. - **Intraday**. Multiple execution times within a day. - **Daily**, **Weekly**, or **Monthly** allow for fine-grained control of the frequency schedule
Start Date on Server	For one-time (Frequency = Once) execution, type in a date, or select the Date Chooser to enter a date. In the Date Chooser, select Now to enter the current day.
Start Time on Server	The default is midnight (00:00:00). Type in a time of day, or select the Date Chooser to enter a time. In the Date Chooser, select Now to enter the current time.
Schedule Options	- For **Intraday**, select a period of hours, minutes or seconds. Optionally enable the start and end hours. - For **Daily**, select *Every Day* (default), *Every Weekday* (M-F), or *Every n days* - For **Weekly**, select one or more days of the week. For example, select Monday to execute every Monday. Select Monday and Thursday to execute every Monday and Thursday. Optionally, specify a weekly execution frequency of 1 to n, where n schedules every 2nd, 3rd, or nth week, such as 12th week. - For **Monthly**, select a specific day, or select the first, second, third, fourth, or last day of the month. For example, select Last Monday to execute on the last Monday of the month. **Tip:** To execute on the last day of each month, enter a Day of **31**. The correct last day of the month is adjusted. Select the months individually, if desired. By default, all months are selected.
Run Options	Select from the following options: - **Skip if previous still running** (*default*). Does not schedule another process instance if the previous instance did not complete. - **Terminate previous if still running**. Terminates a previously scheduled execution prior to starting a new instance. - **Always create new process** . Schedules a new process execution regardless of the status of any prior executions.

Execution Schedule	
Service Type	Choose whether the service is BPEL or IPD-based.
Service Name/ Select Service	If the service type is BPEL, type in a service name. Open the **Service Definitions** list to view names, then type in just the service name (without any extensions), such as `myScheduledService`. If the service type is IPD, select one of the services in the picklist.
Input Document/ POST Body	If the process is implemented using Process Developer and of BPEL type, follow either of the below steps: - If the process implements a SOAP interface, enter the SOAP input message data contained within the SOAP Body Element, that is, the XML input message used to invoke the process. The data must conform to the WSDL interface of the process. - If the process implements a REST interface, you first need to create a SOAP wrapper for the REST service. Then, follow the instructions above. If you use Process Designer to implement the process and the process is of the Informatica Process Definition (IPD) type, enter the JSON body of the POST message used to invoke the process.

After you enter this information, Process Server creates a new scheduled process and you see that it appears in the Scheduled Process page.

See also *Managing and Modifying Execution Schedules*.

Managing and Modifying Execution Schedules

Use the icon, check box, and drop-down list of the Process Schedules list to manage the list.

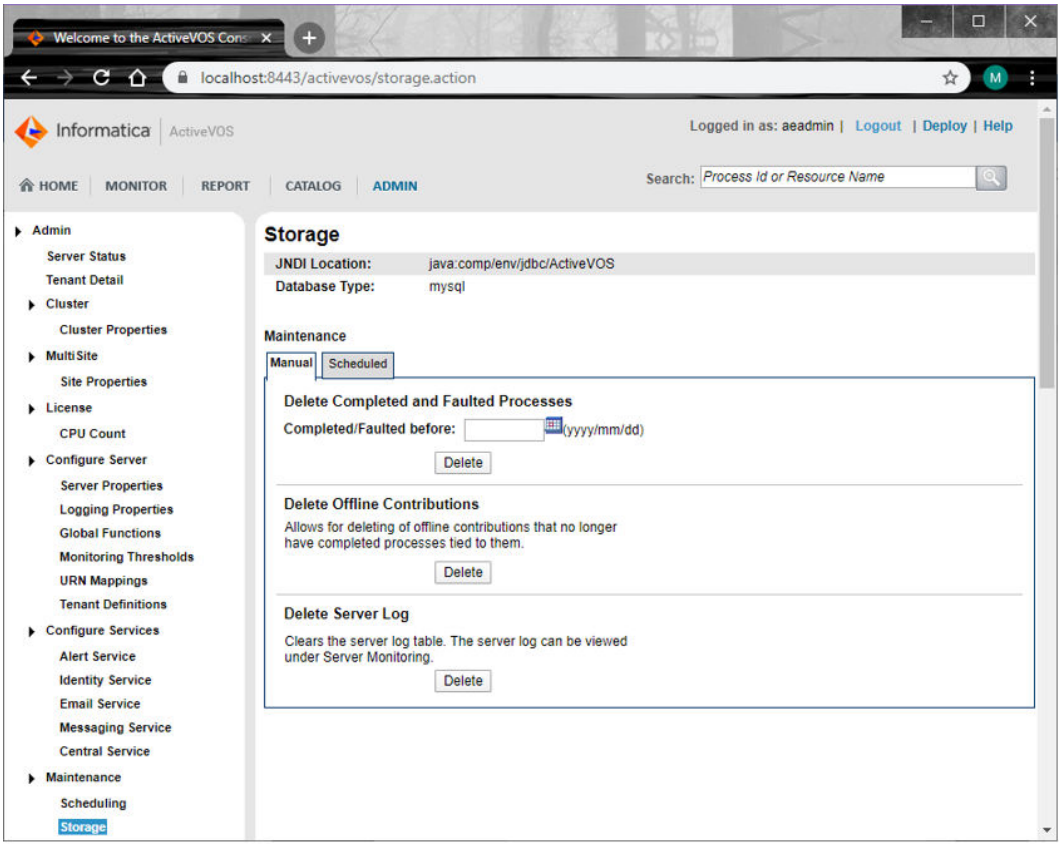
Field	Explanation
Pencil icon (Edit)	Select the pencil icon edit the schedule
Select/Select All	Select an individual execution schedule or select the Select All check box to run, duplicate, delete, enable, or disable the schedule
Run Now	Execute the selected processes
Duplicate	Duplicate selected schedules. Initially the new schedules are disabled.
Delete	Delete the selected execution schedules, including all details
Enable/Disable	Enable or disable the selected execution schedules. If you enable a schedule, the process executes at the next execution time. **Tip:** A disabled schedule may still be executed immediately (Run Now), duplicated, or deleted.

Storage

Note: You only see this command if you have selected an agent.

Process Server includes persistent storage based on the database settings you configured during installation. You must configure a database before running the engine.

The Storage page displays database configuration properties and allows you to maintain the database.



The displayed properties are:

Property	Explanation
JNDI Location	The Java Naming and Directory Interface (JNDI) context that specifies where to look for the database. For example,`jdbc/ActiveVOS`
Database Type	The type such as `mysql`. For a list of supported types, see the "Prerequisites" topic in *Process Server Install, Configure, and Deploy* in this help.
User Name	Username, if required, for Process Server's access to the database

Multisite Configuration

The prerequisites for using Multisite Clustering include the following:

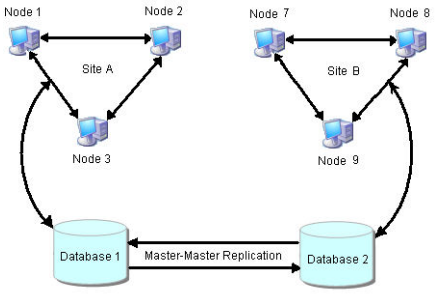
- This feature requires a Multisite license. Be sure to obtain a license to enable this feature in Process Server. The server configuration instructions describe how to enter your license. Enter the license into one server in your Multisite environment.
- You must use either an Oracle or MySQL database. Currently Multisite supports Oracle Database 11g and MySql.
- Use of Multisite requires that the Process Server Eventing Service is disabled. Eventing is disabled in a Multisite license.

- Follow all instructions below to ensure successful Multisite configuration.

What is Multisite?

Process Server supports a site configuration, which allows a cluster of process servers to replicate its database to other databases serving other sites. Each site has a cluster of Process Servers (one or more servers) with a single persistent database, and the site operates independently of other sites. All process instances are available for execution at each site. In the event of a server failure, a process can start execution on one site and complete on a different site. In addition, most Process Server configuration properties are replicated from site to site. This means you can configure server properties in one site, and they are automatically copied to another site's administration console.

The following illustration shows a sample Multisite environment. The database setup is a master-master replication, as shown below. This technique allows you to replicate data in tables across separate databases.



Oracle and MySQL

The procedures you follow when you are using Oracle differ from those you use when using MySQL. Use the following links to jump to the description for your database.

- Oracle
- MySQL

Following these descriptions is a section describing "Pausing Replication When Performing Database Updates. This section applies to Oracle and MySQL.

Database Replication Setup and MultiSite Configuration

The MultiSite page does not show any site information until you set up and configure databases for replication on each site.

For set up and configuration details, refer to the *Process Server MultiSite Configuration* section within the *Sever Install, Configure, and Deploy* help.

MultiSite Page Details

The MultiSite page contains the following information.

Field	Explanation
Name	Name added to the Site Properties page of the Process Console associated with the site. You can update site properties for the current site; however, you cannot update one site's properties on another site's Process Server. Connect to a site's Process Console using this name.
Status	Status values are as follows: - Available—the site is sending heartbeats. If a site was unavailable or failed over, it changes to available when it begins to communicate. - Unavailable—the site is not communicating. It has exceeded the maximum number of missed heartbeats set in the Site Properties page. An alert is automatically triggered, and sent to the Monitor Alert Service, if the service is deployed and running. For details, see *Monitoring Thresholds, Properties, and Monitor Alert Service*. The site is not immediately failed over. You can manually fail over the site, as described in Failing Over a Site Manually. - Failed Over—you have manually failed over an unavailable site, as described in Failing Over a Site Manually. - Incompatible—the site is using a software version that does not match the other sites.
Id	A numeric value in the range of 0 to 127 which uniquely identifies the site (There is a limit of 128 sites). Process Server generates an Id after database replication is configured.
Description	Description of the site as it appears in the Site Properties page.

Failing Over a Site Manually

There may be times when inter-site communications go down but sites are still running. In this case, a site is considered unavailable.

When a site becomes unavailable due to a communication error, it is not immediately failed over to another site. Instead, Process Server sets the site status to Unavailable, triggers an alert, and provides a Fail Over action for you to take, if desired.

Once a site has failed over, a new state called Failed Over is displayed, and the Fail Over button is removed.

When the unavailable/failed-over site begins communicating with the other sites (sends a heartbeat), the site automatically becomes available.

Monitor Alert Service

When a site becomes unavailable, an alert is automatically triggered. If Process Server finds a deployed Monitor Alert Service, the service will run. For details, see *Monitoring Thresholds, Properties, and Monitor Alert Service*.

Site Properties

You can configure the following properties for the site associated with this Process Console. Each Site Properties page must be configured on a Process Server in its own site.

Field	Explanation
Name	Type in a short name with no spaces. This name appears in the MultiSite page of each Process Console in all the sites. It also appears in the title area of the Process Console. If no name is provided, the host name defined in the Site Service URL is used.
Description	Optional. Type in a short description of the site. The description appears in the MultiSite page of each Process Console in all the sites.
Id	Id assigned to the site by Process Server; this is used in internal communication.
Process Console URL	Type in the address of the Process Console for this site. The URL is defined during configuration and deployment of the server. The URL is the address that other sites use for a link from within the Process Console.
Site Service URL	Define how the site should be contacted for multi-site communication. The address will most likely contain the address of the load balancer installed in front of the servers.
WS-Security Enabled	(Replicated). By default, WS-Security is not enabled. Be sure to configure WS-Security on each node in each site before you enable it for MultiSite. For details on configuring WS-Security, refer to *Configuring Process Server for WS-Security*. within the *Process Server* help. If your site environment has another security implementation, you can leave WS-Security disabled.
Allow Optimistic Deployment of User Contributions	(Replicated). If selected, site locking is not performed when BPR contributions are deployed. This means that you are assuming that the same BPR contribution is not deployed from two sites at exactly the same time, and each site does not need to be locked. Enabling this is useful in a multi-tenant environment because it allows for faster deployments. This option does not apply to system deployments. Sites are always locked for these deployments.
Heartbeat Interval	(Replicated). Set the frequency in seconds to poll the site for its availability. The default is 15.
Max Missed Heartbeats	(Replicated). Set the maximum number of missed heartbeats before assuming the site is unavailable. If a site does not respond, the site status changes to Unavailable, and database replication cannot occur. The default is four.
Expected Database Latency	(Replicated). Set the latency in seconds expected for database replication to occur. The default is five seconds. This value serves as a guideline for anticipated database replication. This setting is necessary for the coordination of site to site communications with database replication. Site to site communication is typically completed in milliseconds while database replication varies, depending on your hardware and other environmental factors. A database latency of five seconds attempts to synchronize messages sent via site communications with database replication every five seconds until synchronization is either successful or the until the database latency timeout is reached.
Database Latency Timeout	(Replicated). Set the latency timeout value in seconds. The default is 300. This setting indicates that changes made to one site's database instance have not been replicated to another site within the latency period. If this value is reached, a critical storage error is written to the application Process Server. You can also configure a server property to monitor storage exceptions. For details, see *Monitoring Thresholds, Properties, and Monitor Alert Service*.

Site Independent (Non-Replicated) Data and Configuration

All database information is replicated between sites except the following:

- URL/URN mappings
- Meta-data about the site, such as communication details, database name, scheduled maintenance
- Data from the `AeLock` table, which is a local lock table
- Engine monitoring for engines in each site's cluster

Also, starting and stopping a server is confined to one server only.

For process scheduling, a single node is designated from all known sites to be responsible for scheduling. Schedule changes are broadcast between sites so that the scheduler stays in sync with schedule changes made on other sites.

Configuring File Locations for Identity Services and Global Functions

A few configuration tasks require that you specify a file location for certain files. Process Server allows you to specify an absolute path or to specify the location using an environment variable (or similar). In a MultiSite environment, you must ensure that required files have valid path locations for each site. This may mean that you copy a file to each site location or to store a file in a centralized location. The files can include:

- File-based Identity service.
- LDAP Identity Service Trusted keystore file.
- Global Custom functions.

MultiTenancy

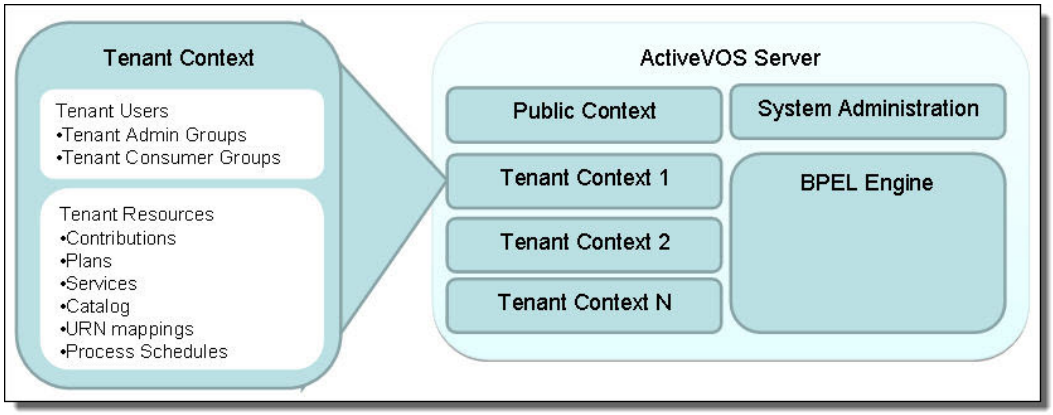
This feature requires a Multi-Tenant license. MultiTenancy is not available for the Process Server embedded in Process Developer.

As a prerequisite to this feature, the Process Server must be configured with security roles. Refer to the *Process Server* help for details on security setup.

What is MultiTenancy

Multitenancy allows separate organizational groups to share a Process Server while maintaining privacy in their business processes. The Process Server administrator creates a tenant profile, which contains tenant administrators, users, and other details. Each tenant develops, deploys and manages their own set of private contributions. All tenants share the same server configuration, but they have access to a subset of configuration and maintenance features for their own use.

The following illustration shows an overview of the multitenant Process Server environment.



Step 1: Setting Up Identity Service Groups with Security Roles

As a first step, the Identity Service administrator in your organization must assign the tenant security role, `abTenantAdmin`, to the group who will administer the tenant context on the Process Server and will deploy contributions from Process Developer. This role gives an authenticated administrative user access to tenant-related pages of the Process Console and the authorization to deploy contributions into the tenant context.

In addition, users who will instantiate processes must have the `abServiceConsumer` role.

Example Security Role Setup

Identity Service Groups/Members	Role and Group Assignments
Tenant101AdminGroup	`abTenantAdminabTaskClient`
- tenant101Admin1 - tenant101Admin2 - tenant101Developer1	Tenant101AdminGroup `abTenantAdmin` abTaskClient
Tenant101UserGroup	`abSeviceConsumerabTaskClient`
- tenant101ProcessInitiator1 - tenant101TaskUser2	Tenant101UserGroup `abSeviceConsumer` `abTaskClient`

After the tenant admin role is assigned to user groups, and you set up a tenant, a user with `abTenantAdmin` privileges can log into the tenant context of Process Console as shown in the following example:

```
http://localhost:8080/activevos/tenant101
```

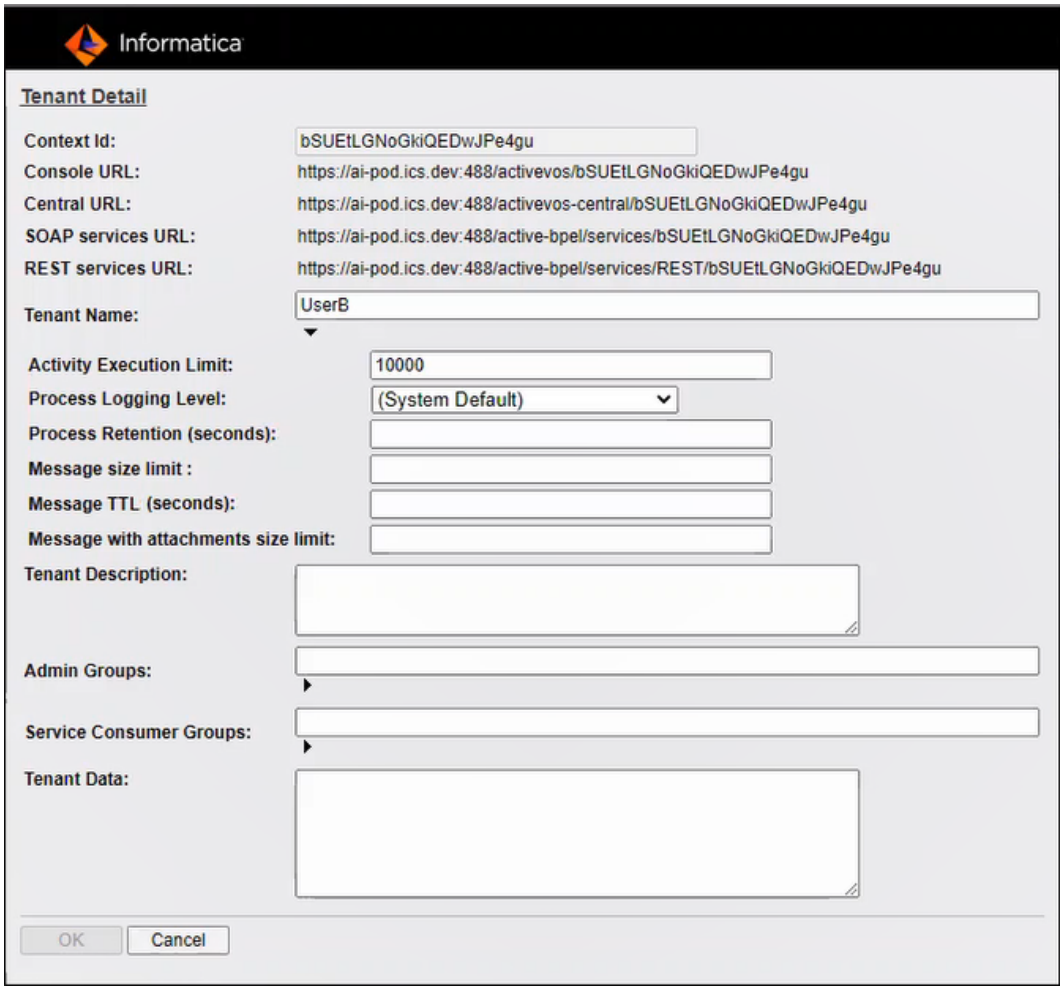
Note that access to `http://localhost:8080/activevos` is denied to any user who does not have the `abAdmin` role.

Step 2: Setting Up a Tenant Definition

Begin with the following steps:

1. Login to the Process Console as the System Administrator.
2. From the Admin page select Tenant Definitions.
3. On the Tenant Definitions page, select **Create Tenant** button.

Here is the Create Tenant Dialog:



Type or enter the following in this dialog::

- **Context Id**. This Id is used in the URL to login to the Process Server, deploy contributions, and access services. For example, a tenant administrator would login to the Process Console at the following URL: `http://localhost:8080/activevos/tenant101`
- **Console URL**. The URL that is used to invoke the tenant's Process Console.
- **Central URL**. The URL that is used to invoke the tenant's Process Central.
- **SOAP services URL**. The URL to which a SOAP request is sent.
- **REST services URL**. The URL to which a REST request is sent.
- **Tenant Name**. This descriptive name is for system administrator use.
- **Activity Execution Limit**. Set a value for the maximum number of activities that can execute In order to avoid infinite loops and resource draining processes. The default value is approximately 10,000. When this maximum is reached, the process is suspended. Suspended processes must be resumed manually. The processes count is reset to zero when it is reset. You can get around this limitation by using an agent to run your process.
- **Process Logging Level**. By default, Process Server generates an execution log for running processes. You can view or download an execution log for a running or completed process. An execution log provides

start and end times for activity execution and helps you troubleshoot faulted processes. The logging levels are:

- Execution (default)—All execution statements are logged, except for *Will Not Execute* statements. Using this setting can greatly decrease the size of the log file.
- Execution with Data—All execution statements are logged, except for *Will Not Execute* statements, but including variable, expression, and partner link data. Using this setting can increase the size of the log file.
- Execution with Service Data—All execution and fault information, as well as some WSIO activty information. For execution information, information such as deadpath states, terminations, ready-to-execute and the like. For WSIO, this includes invokes, picks, and receives. WSIO information related to data assignment and changes is excluded.
- None—You can disable logging to enhance engine performance.
- Full—All execution statements are logged, including the *Will Not Execute* statements for deadpath activities. For example, all fault handling statements that are not executed are logged.

- **Process Retention (seconds)**. Specify how many second to keep completed and faulted processes are retained before they are deleted.
- **Message TTL (seconds)**. Specifies the amount of time between the time a message reaches the Process Server and the time when it is about to get dispatched to the target process. Default is 24 hours.
  When you configure the message time to live (TTL) period and invoke a process by using the REST or SOAP endpoint, if the HTTP message is not dispatched within the message TTL period, the request fails. You see the HTTP error `503 Service Unavailable` with the following error message:

  ```
  Message discarded having exceeded the defined TTL.
  ```

  The message TTL is applicable to the endpoints with the following entries:

  - `/active-bpel/services/`
  - `/active-bpel/public/rt/`
  - `/active-bpel/rt/`
  - `/active-bpel/public/soap/`
  - `/active-bpel/soap/`
  - `/active-bpel/odata/repository/v4/OdataRepository/Execute`
  - `/active-bpel/tf/`
  - `/activebpel-cert/services/`
  - `/activebpel-cert/rt/`
  - `/activebpel-cert/soap/`
  - `/activebpel-cert/odata/repository/v4/OdataRepository/Execute`
  - `/process-engine/rt/`
  - `/process-engine/public/rt/`
  - `/process-engine/services/`
  - `/process-engine/public/soap/`
  - `/process-engine/soap/`

  Note: If the message TTL is not configured for the tenant, the message TTL configured for the Process Server applies.
- **Message with attachments size limit**. Sets the maximum size of the payload for a multi-part message that has attachments.

If you use the Informatica Cloud Server, you can set a maximum message payload size of 5 MB. The maximum size for attachments is also 5 MB.

If you use a Secure Agent, the default message payload size is 5 MB and the default attachment size is 5 MB. You can change the default sizes to set higher or lower values.

- **Tenant Description**. These details are for system administrator use.
- **Admin Group**: Security Roles:

  - `abTenantAdmin` (Required for the group and each user in the group)
  - abTaskClient (Required for each user in the group for access to Process Central)
  - Not needed: `abDeployer` or `abServiceConsumer`
  - `abDeveloper` (Allows remote debugging from Process Developer. Alternately, tenant admins can perform debugging in the Process Console.)

  Privileges associated with `abTenantAdmin`:

  - Login access to the tenant context and control over tenant context (described in "Tenant Administration for Tenant Administrators Deployers" on page 97)
  - Deploy contributions into the tenant context from Process Developer
  - Access Process Server Web Services
- **Service Consumer Groups**: Security Roles:

  - `abServiceConsumer` (Required for the group and each user in the group)
  - `abTaskClient` (Required for each user in the group for access to Process Central)

  Privileges: Start processes, work on tasks in Process Central within the tenant context
- **Tenant Data**. This data is not currently used.

Tenant Administration for System Administrators

Any user with an `abAdmin` security role can login to the Process Console or can login to any tenant context.

For example, a Process Server administrator can log into:

```
http://localhost:8080/activevos/tenant101
```

When a Process Server administrator logs into a tenant context, the administrator can read and execute exactly the same Process Console pages and features as a user in a tenant Admin Group.

When a Process Server administrator logs in without using a tenant context, the administrator sees tenant-related data on many Process Server pages and can filter the listings by tenant, including:

- Active Processes
- Contributions
- Process Definitions
- Service Definitions
- Catalog Resources

Only system administrators are allowed access to Process Server administration tasks.

To manage memory allocation on a tenant basis, create a dispatch service for each tenant.

Tenant Administration for Tenant Administrators Deployers

Any user defined in a tenant Admin Group can log into the tenant context of the Process Console. For example, a context looks like:

```
http://localhost:8080/activevos/tenant101
```

Access to other services include the following URLs:

```
http://localhost:8080/activevos-central/<TENANT Context Id>
http://localhost:8080/active-bpel/services/<TENANT Context Id>/serviceName
http://localhost:8080/active-bpel/catalog/<TENANT Context Id>/locationPath
```

To create a URN mapping, the tenant context can be added:

```
http://localhost:8080/active-bpel/services/<TENANT Context Id>${urn.4}
```

A tenant administrator has access to the following within the tenant context:

- Deploy BPRs into their context
- View Catalog Resources. There is full control for tenant listings and read-only view of public listings.
- Active Processes
- Active Tasks
- Deployment Logs
- URN Mappings
- Scheduled Processes
- Reports
- Server Status

Tenant Access to Public Resources

In a MultiTenant environment, each tenant has access to its own resources, and in addition, a tenant has read and execute access to public resources that a system administrator has added to the server. This means that a process can be deployed to a tenant context and invoke another process in the public context or use resources in the public context.

By default, there is one tenant context, `$public`, that contains common, system-wide deployments.

- Public resources can include:
- Contributions including processes and related resources
- Catalog resources, such as a common XSL file for the doXSLTransform function that might be used in tenant processes
- Services created from system services, such as a Shell Command
- Process Central configuration files and properties
- Global custom functions

Any requests without a tenant context id is assumed to be deployed as a public service.