# Table of Contents

**JAMS Installation Guide**  
2-25  
**Agents in JAMS**  
26-27  
**OpenVMS Agent**  
28-31  
**JAMS Agent for System i**  
32  
**JAMS Agent for Windows**  
33  
**JAMS Agentd for Linux/Unix**  
34-36  
**zOS**  
37-39  
**Integration Packs**  
40  
**Banner**  
41-44  
**Dynamics AX**  
45-51  
**JD Edwards**  
52-53  
**Informatica Cloud**  
54-55  
**Micro Focus**  
56-58  
**Microsoft SCOM**  
59-63  
**NeoBatch**  
64-66  
**Netezza**  
67-70  
**Paladin**  
71-91  
**PeopleSoft**  
92-101  
**SAS**  
102  
**SAP Execution Methods**  
103-108  
**SAP Business Objects Data Services**  
109-110  
**Symitar**  
111-119  
**zOS**  
120-122  
**Oracle EBS Integration**  
123  
**Automating Oracle EBS Jobs**  
124-125  
**Configuring The OracleStoredProc Execution Method**  
126
JAMS Installation Guide

NOTE: Uninstall features are present in the Installation Wizard. Unchecking component boxes in the Add-Ons section will uninstall those components.

Installed Components

The JAMS environment incorporates several key components that work together to create a unique and powerful scheduling system.

**JAMS Client**

The JAMS Client provides the primary user controls for JAMS and includes:

- Windows-based GUI (JAMS Client)
- PowerShell Cmdlets
- JAMS PowerShell provider
- .NET-based Class Library
- JAMS.exe command line tool

**JAMS Scheduler for Windows**

This is essentially the heart of the JAMS system. The scheduler provides background services that maintain, schedule and execute JAMS Jobs.

**JAMS Agent for Windows**

JAMS Agents provide background services that execute JAMS Jobs under the direction of a JAMS Scheduler located on a different machine.

**JAMS Agent for OpenVMS**

The JAMS Agent for OpenVMS provides background processes that execute JAMS Jobs under the direction of a JAMS Scheduler located on a different machine.

**JAMS Web Client / REST API**

Optional. The JAMS Web Client, when installed, allows users to run and manage the automation environment from any web-connected device.

Preparing for the JAMS Installation

<table>
<thead>
<tr>
<th>JAMS Scheduler</th>
<th>JAMS Agent</th>
<th>JAMS Client</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software Prerequisites</strong></td>
<td><strong>.Net Framework v.4.6 (or higher)</strong></td>
<td>.Net Framework v.4.6 (or higher)</td>
</tr>
<tr>
<td></td>
<td>Microsoft Message Queue (installed with JAMS)</td>
<td>.Net Framework v.4.6 (or higher)</td>
</tr>
<tr>
<td></td>
<td>Microsoft SQL Server 2012 or newer</td>
<td><strong>Hardware Server</strong> (minimum)</td>
</tr>
<tr>
<td></td>
<td><strong>Processor:</strong> Dual Core ~2.5 GHZ (x86 or X64)</td>
<td><strong>Processor:</strong> 1 GHZ (x86 or x64)</td>
</tr>
<tr>
<td></td>
<td><strong>Memory:</strong> 8 GB</td>
<td><strong>Memory:</strong> 1 GB</td>
</tr>
<tr>
<td></td>
<td><strong>Disk Space:</strong> 10 GB, in addition to the disk space required by the OS</td>
<td><strong>Disk Space:</strong> 75 MB</td>
</tr>
<tr>
<td></td>
<td><strong>Processor:</strong> 1 GHZ (x86 or X64)</td>
<td><strong>Memory:</strong> 1 GB</td>
</tr>
<tr>
<td></td>
<td><strong>Memory:</strong> 300 MB</td>
<td><strong>Disk Space:</strong> 300 MB</td>
</tr>
</tbody>
</table>

©2020 HelpSystems, LLC. All Rights Reserved.
### Hardware Server

<table>
<thead>
<tr>
<th>(50,000 to 100,000 Jobs per Day)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong>: 8 Core ~3.2 GHZ (x86 or X64)</td>
</tr>
<tr>
<td><strong>Memory</strong>: 32+ GB</td>
</tr>
<tr>
<td><strong>Disk Space</strong>: 1+TB</td>
</tr>
<tr>
<td>Recommended: A separate SQL Cluster for the JAMS SQL Back-end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(&gt;100,000 jobs per day)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong>: 16 Cores, ~3.2GHZ</td>
</tr>
<tr>
<td><strong>Memory</strong>: 64+ GB</td>
</tr>
<tr>
<td><strong>Disk Space</strong>: 1+ TB</td>
</tr>
<tr>
<td>One server hosting the JAMS Scheduler</td>
</tr>
<tr>
<td>One server reserved for the SQL Server with multiple high-speed SCSI drives</td>
</tr>
</tbody>
</table>

### Hardware Server

<table>
<thead>
<tr>
<th>(&gt;100,000 jobs per day)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong>: 16 Cores, ~3.2GHZ</td>
</tr>
<tr>
<td><strong>Memory</strong>: 64+ GB</td>
</tr>
<tr>
<td><strong>Disk Space</strong>: 1+ TB</td>
</tr>
<tr>
<td>One server hosting the JAMS Scheduler</td>
</tr>
<tr>
<td>One server reserved for the SQL Server with multiple high-speed SCSI drives</td>
</tr>
</tbody>
</table>

### Permissions

To install JAMS, ensure that the user performing the installation is logged in as an account with administrator privileges to the application server. In addition, the account must have the ability to create a database on the database server. JAMS creates the database locally by default. Typically, giving the user sysadmin rights takes care of all necessary permissions.

### Prerequisites

Before installing JAMS, please review the following requirements detailing the prerequisite software and minimum hardware requirements.

The JAMS self-extracting installer checks for and installs all prerequisites.

#### .NET Framework v. 4.6

The JAMS Client and Scheduler require .NET Framework v. 4.6 (or higher). Once the JAMS installation begins, it checks if this version of .Net Framework is installed. If v. 4.6 is not installed, the installation will ask users to accept Microsoft’s license agreement. Once the agreement is accepted, the software will then download and install the framework.

**Note**: Installing v.4.6 of the .NET Framework will require a system reboot.

#### .NET Framework v. 4.6

The JAMS Agent requires v.4.6 (or higher) of the .NET Framework.

#### Microsoft Message Queue (MSMQ)

The JAMS Scheduler requires the Microsoft Message Queue (MSMQ). When installing the JAMS Scheduler, the installer will check for the MSMQ and, if it's not installed, will ask you to confirm that you want to install it. For more details, go the **MSMQ** section in this topic.

**Note**: MSMQ is included with Microsoft Windows, but it is optional and not installed by default.

#### Microsoft SQL Server

The JAMS Scheduler requires Microsoft SQL Server 2012 or newer. The SQL Server does not need to be installed on the same machine as the JAMS Scheduler.

If you or your DBA choose to install the JAMS Scheduler component, the installer will first check for a connection to an SQL Server. If the server not installed, the installer will ask to install the SQL Server 2017 Express Edition.
You also have the option of using an SQL Server on a different machine. Please note, when the installer checks for the SQL Server, it only looks for the default instance names of MSSQLSERVER and SQLEXPRESS.

If you have a different instance preference, decline the SQL Express installation and specify your preferred instance when the JAMS installer asks for the SQL server name and instance.

**NOTE:** To run SQLStoredProcedure Execution Method Jobs, Shared Management Objects for your version of SQL will be needed. Shared Management Objects can be found in the appropriate Microsoft Feature Pack version, or in the install for SQL Management Studio.

**NOTE:** To run SSISDirect Execution Method Jobs, the Client Tools Backwards Compatibility and Client Tools SDK Shared Features are required.

**PowerShell V3 and Up**

The JAMS Client component includes a PowerShell Snap-In that contains a number of JAMS cmdlets and a JAMS Provider. PowerShell is not required, but in order to use the Snap-In you must install PowerShell before installing JAMS. If you happened to install PowerShell after installing the JAMS Client, just reinstall the JAMS Client to pick up the JAMS Snap-In.

**Note:** JAMS supports PowerShell v. 2 and higher through the use of V6.X Agents. A V6.X Agent may be connected to a JAMS V7.X Scheduler Server, allowing JAMS V7 to run PowerShell V2.

**Hardware Server Specifications for the JAMS Scheduler**

- Processor: ~2.5 GHZ (x86 or x64)
- Memory: 8 GB
- Disk Space: 10 GB, in addition to the disk space required by the OS.

These are the minimum system requirements that can support running thousands of jobs per day. Heavy loads, for example, over 100,000 jobs per day, will require more resources.

For database planning, the JAMS test lab runs approximately 1,000 jobs per day using one instance of JAMS. Retaining 30 days of information generates a database approximately 110 Mb in size.

**For sites running over 100,000 Jobs per day**

Based upon customer implementations where more than 100,000 jobs are executed each day, the JAMS installation should consist of:

- Two Windows servers, each with Dual Quad 2.0 GHz CPUs and 32 GB of RAM (or more).
- Specifically, one of the above servers hosting the JAMS Scheduler and the second server reserved for the SQL Server with multiple high-speed SCSI drives.

**Hardware Server Specs for JAMS Agents**

- Processor: 1 GHZ (x86 or x64)
- Memory: 1 GB
- Disk Space: 75 MB

**Hardware Server Specs for the JAMS Client**

- Processor: 1 GHZ (x86 or x64)
- Memory: 1 GB
- Disk Space: 300 MB

**Supporting JAMS in a Virtual Environment**

MVP Systems Software, Inc. supports customers running JAMS on any supported operating system in a virtualized environment, such as VMware and HyperV.

**Installing JAMS**

Follow the steps below to install JAMS on a Windows computer.
1. Log on to Windows using an account with administrator privileges. (Typically, using an account with sysadmin rights takes care of all necessary permissions.)

**NOTE:** Users must also have the ability to create a database on the database server. The default database server is the local machine.

2. Right click the SetupJAMS executable and select Run as Administrator to start the JAMS installer.

3. On the Welcome step, verify this installation is the proper edition and version. Click the Next button.

4. On the License Agreement step, review and accept the JAMS license agreement. Click the Next button.

5. On the Core Components step, select the JAMS Components to install. If this is a new installation where no Scheduler exists, the JAMS Client and JAMS Scheduler should be installed at minimum. When finished, click the Next button.

6. On the Integration Packs step, select the packs to install. Note that Integration Packs are not required to install JAMS. Click the Next button.

7. On the Contact Information step, enter the information that will be used when working with the JAMS technical team. Click the Next button.

8. On the SQL step, enter the information that will be used to create the JAMS database, as well as the authentication used to connect to the JAMS database. Click the Next button.

**NOTE:** The JAMS Scheduler Service will not start until the JAMS database is created.

**NOTE:** Users should NEVER attempt to manually create the JAMS database.

9. On the Locations step, define the preferred target directory or accept the default for JAMS and the JAMS database files. Click the Next button.

**NOTE:** In an upgrade, the target directory cannot be modified.

10. On the JAMS Sample Jobs step, decide whether the sample jobs should be installed. If installing sample jobs, define a JAMS user to run the sample jobs. Click the Next button.

11. On the Web Client Details step, select where the web client should be installed.

12. You may be presented with a page that asks you to confirm that you want to install MSMQ.

13. In addition, you may be presented with a page that asks if you want to download and install the SQL Server Express Edition.

14. On the Finish page click the Finish button to begin the installation.

**JAMS Client Installation Components**

The JAMS Client installation includes the following items:

- A Start Menu shortcut to JAMSWin.exe
- The JAMSShr.dll Class Library
- The JAMS PowerShell Snap-In (only if PowerShell is installed at the time of the JAMS Client installation)
- The JAMS.exe command line tool
- JAMS Help files

**JAMS Scheduler Services Installation**

The JAMS Scheduler installation also contains other important JAMS components including:

- JAMS Scheduler Services
- JAMS Server Services
- JAMS Executor Services

**Note:** If this is the initial installation of the JAMS Scheduler, you will be prompted for additional information when the installer configures the JAMS Database.

**JAMS Command Line Environment**

In addition to the JAMS Client GUI, the Powershell SnapIn, and the .NET Controls, JAMS offers a command line interpreter to manage your scheduling environment.

When the JAMS Client installs, it also includes a command line tool (JAMS.EXE)

1. When started from the command line, a JAMS> prompt appears.
2. Type HELP to reveal all the supporting commands. Type HELP COMMAND to get additional assistance for a particular command (e.g., HELP EXPORT JOB).
Your first action establishes the connection to the JAMS ServerName.

Creating the JAMS Database

The JAMS Installer will create the database, unless users specify otherwise. The database should always be created by JAMS. If the database was not created on installation, users may manually start the JAMS Database Creation Wizard.

**Note**: Installing JAMS without a database won’t affect the JAMS installation, but JAMS cannot function without a database. To reinstall the JAMS Database, start the wizard as described in the **Troubleshooting** subsection below.

Troubleshooting: Starting the JAMS Database Creation Wizard

If you encounter problems with the JAMS Database, start the JAMS database creation wizard by following these four steps:

1. Open a command window (PowerShell or CMD.EXE)
2. Set your default directory to the JAMS Scheduler default directory (`C:\Program Files\MVPSI\JAMS\Scheduler`).
3. If the file `Common.config` file exists, delete it.
4. Enter the command: `JAMSDBA INSTALL`

Support Contract Information

Support Contact Information page requests information to be sent to JAMS technical support in the event of a failure. This information includes the:

- Company Name
- Contact Name
- Phone Number
- E-Mail Address
- SMTP Server

**Note**: Populating this page is optional, but entering accurate information can help JAMS support staff improve its support capabilities. If you’re unsure about the information, leave it blank and enter it at a later time.

SQL Server Page

On this page you will specify the name of the SQL server machine and the SQL Instance. If the SQL Server is installed on the local machine, the default will be (local). If the server you want isn’t listed in the pull-down control, you can enter the name and instance manually.

Database Name

This is the name of the database that will be created. The default is JAMS. However, you will need to change this to a unique name if, for example, you are running the JAMS Scheduler on two different machines but are sharing the same SQL Server machine.
Authentication

Select the type of authentication to use when creating the JAMS database. If you select SQL Server Authentication, you must also supply a database user name and password.

Location of JAMS Database Files Page

The JAMS installation allows users to specify the location of JAMS database files.

The JAMS database is divided into three data files and a log file. These paths are on the database server machine, which may not be the same machine where JAMS is being installed.

Primary File

The Primary data file holds most of the database tables. These tables store the definitions of JAMS objects. Data is inserted and deleted only when JAMS objects are created or deleted.

Volatile File

The Volatile data file contains database tables that have records inserted and deleted whenever a task executes. This could equate to hundreds or thousands of inserts/deletes per day.

History Data File

The History data file includes historical database tables. A record is inserted into this file every time a task is executed, depending upon how much history you choose to keep. If not controlled, the data file can easily balloon in size.

Database Logs

The database log file holds transaction information that is used to recover the database in case of a failure. If possible, you should place the database log file on a separate disk from the other database files.

Default Directories

The JAMS installation allows users to specify the location of JAMS temporary files and log files. These paths must be on the local machine.

Temporary Files

In most cases, JAMS creates a temporary script file when a Job is executed. These temporary files are created in the directory specified in this file. You can change this directory using the Configuration shortcut option in the JAMS Client.

Job Log files

When a Job is executed, JAMS keeps a log of the run. You can specify the location of the log file from the Job Definition or in the Job's Folder Definition. If the location isn't specified in one of those two places, it defaults to the value specified in the Job Log file. You can change this directory using the Configuration shortcut option in the JAMS Client.

Account Information

When the JAMS database is initially loaded, a folder named JAMS is created along with a number of Jobs. There are also a number of sample Jobs installed. These Jobs cannot run unless they have an Execute As user credential to run. You will be presented with a dialog requesting a user name and password for these Jobs.

If you don't enter a user name and password, it can be added at a later time.

NOTE: To run jobs in JAMS, an Execute As user MUST be specified.

Installing the JAMS Web Client (REST API)

Prerequisites

- JAMS V7.0 or higher
- Windows Server 2012 R2 or higher
- .NET 4.6 or higher
- IIS 8 or higher
- Edge, Chrome, or Firefox recommended for the best experience

IIS Installation
Begin by enabling the Web Server IIS Role through the Server Configuration screen.

Ensure the following components are enabled within Application Development:
JAMS Installation

Download the JAMS Installer. During the Installation, ensure the “JAMS Web Client” component is selected.

Once the JAMS Installation is complete, open the Common.Config file for the Web Client, located at C:\Program
Edit the `Common.Config` file as necessary for the given environment:

1. Verify that the connection string value matches the desired server.
2. If the JAMS Site is installed within a Domain, set the `ContextType` value to “Domain” and `ContextName` value to the domain itself. *(In the image below, the domain is ‘JAMS’)*
3. If the JAMS Site is not installed within a domain, set the `ContextType` value to “Machine” and `ContextName` value to the name of the machine.
4. If the JAMS REST API/Web Client is installed on a different server than the JAMS Scheduler Server, set the `SchedulerName` value and `SchedulerPort` value accordingly. *(If the scheduler is not on another server, setting values for SchedulerName and SchedulerPort will cause issues.)*

5. If the JAMS REST API/Web Client is installed on a different server than the JAMS Scheduler Server AND a failover is not configured, the key `<add key="EnableRemoteAccess" value="true" />` must be added to the Common.config file on the JAMS Server, located by default at C:\Program Files\MVPSI\JAMS\Scheduler\Common.config. This will open port 4773, allowing the web client to communicate with the scheduler. In an environment with a Failover configured, port 4773 will already be open. After configuring the JAMS Server common.config file, restart the JAMS Server service.
6. If the JAMS REST API/Web Client is installed on a different server than the JAMS Scheduler Server, the JAMS fat client must be installed on the web server. The client can be downloaded [here](#). The Web Server and Scheduler Server must be running the same version of the fat client.
7. If the JAMS REST API/Web Client is installed on a different server than the JAMS Scheduler Server, users must grant the Web Server full permissions to the jamsrequests Queue on the Scheduler Server. Follow our article to grant access to the MSMQ, using the naming format MyWebServer$.

### App Pool Settings

Add the user running the App Pool to the JAMSApp Role on the JAMS Database created during the initial installation. This will give proper permissions to the JAMS Database.

**NOTE:** It may be necessary to give the DefaultAppPool the JAMSApp Role in SQL. Users are recommended to run DefaultAppPool under LocalSystem for local machine installations. For domain installations, it may be necessary to run the DefaultAppPool as NetworkService.

If your JAMS services are running as a domain account, it is also recommended to run the DefaultAppPool as that same domain account.

Restart IIS services.

### Documentation

The REST API includes built-in documentation displaying the endpoints and corresponding Headers that can be passed. There are two variants of Documentation that can be accessed based upon your preference:
After Installing JAMS

Here are some additional details you may need to know in order to continue using or evaluating JAMS.

Access Control

By default, NT AUTHORITY\Authenticated Users will have full access to JAMS Objects in new installations. However, in order to work with all JAMS functions you must be part of the administrators group.

**NOTE:** In JAMS V7.0.1367 and later, removing all ACEs on an object behaves the same as Windows would. When all ACEs are removed from an object, only the GrantAdministratorsByPass group will have access to the object. Previously, removing all ACEs from an object would give all Authenticated Users access to that object.

The Server ACL controls who may connect to a specific server. The administrators group retains full access with authenticated users granted limited access.

As an administrator, start the JAMS Client and select the **Access Control** shortcut from the JAMS Menu. From the ACL list, choose the **Server** command to make a server selection. The ACL can then be adjusted to define who should have access to JAMS.

When launching the JAMS Client, you will not initially be a part of the Administrators group. To gain administrator access, click on the JAMS Client icon in the Windows start menu and select the **Run as administrator** command from the pop-up menu. This action will open the JAMS Client with Administrator rights giving you permissions to adjust the ACLs for other users.

Setting an Execute As user

For security purposes, JAMS will not run Jobs without an Execute As user defined. An Execute As user may be defined at the Job or Folder level. To allow all Jobs in the environment to run, users may set an Execute As user on the root folder.

To add the Execute As user, navigate to the Properties tab of the Job or Folder definition. Click Add Property, then select Execute As from the property list and click OK. With the Execute As property added, users will be able to select a JAMS Credential from the drop-down list.

**NOTE:** JAMS Properties that display as greyed-out must be added to the object before their value can be defined.

Server Definition

If you're running the JAMS Client on the same machine as the JAMS Scheduler, the JAMS Client can automatically locate the Scheduler.

However, if the Scheduler is running on a different machine, you must include a server definition to tell the client where to find the Scheduler. In order to add a Server definition, click on **Client Settings** in the upper right corner of the JAMS Client and select **Servers** to open the JAMS Servers dialog.

**Note:** you can have many servers defined within the JAMS Client.
Installing JAMS Integration Packs

JAMS natively supports automation using many leading business applications. “Integration Packs” for these preferred applications are selected using a checkbox during the JAMS installation process. Once an Integration Pack for a product is enabled, users can create, manage, deploy and monitor almost any kind of JAMS Job using the following products:

- Banner
- Informatica Cloud
- J.D. Edwards
- MicroFocus
- Microsoft Dynamics AX
- NeoBatch
- Netezza
- Oracle E-Business Suite
- PeopleSoft
- SAP
- Symitar

When installing JAMS, check the Product Integration Pack you want to include. If you have already installed JAMS without the necessary Integration Pack, re-run the JAMS installer, and check the Integration Packs to install.

**NOTE:** In JAMS upgrades, unchecking component boxes will *uninstall* those components

The JAMSDBA Utility

The JAMSDBA.exe utility is used to manage the JAMS Database and to perform other installation and management tasks. JAMSDBA is a command line utility located in the Scheduler installation directory (`C:\Program Files\MVPSI\JAMS\Scheduler` by default).

When starting the JAMSDBA, you will be presented with a JAMSDBA> prompt. You can enter the command you want to execute or enter HELP to access online help.

You can also start JAMSDBA with a command appending the command you want to execute; for example:

```
JAMSDBA UPDATE/LOG
```

The JAMS Client

With the exception of the PowerShell Client, the JAMS Client can also be deployed with an XCOPY installation. Simply copy the contents of the client directory to the client machine.

**NOTE:** The JAMS Web Client only requires a web browser and access to the configured JAMS Site. MVP recommends using the latest release of Microsoft Edge or Google Chrome to run the JAMS Web Client.

**NOTE:** The PowerShell Snap-In is installed only if PowerShell is already installed. If you install PowerShell after installing JAMS, you must reinstall the JAMS Client to pick up the PowerShell Snap-In.
There is no configuration required for the JAMS Client.

```xml
<subkey name="Shortcut006">
  <property name="Type">4</property>
  <property name="Name">JAMS Jobs</property>
  <property name="Title">JAMS Job Definitions</property>
  <property name="Description" />
  <property name="PromptForKeys">false</property>
  <property name="QueryJobName">*</property>
  <property name="QuerySystemName">JAMS</property>
</subkey>

<JAMSServers>
  <subkey name="Server000">
    <property name="Name">Jimmy</property>
    <property name="Node">jimmy.yourco.com</property>
    <property name="Port">773</property>
    <property name="Prompt">False</property>
  </subkey>
  <subkey name="Server001">
    <property name="Name">Joe</property>
    <property name="Node">joe.yourco.com</property>
    <property name="Port">773</property>
    <property name="Prompt">False</property>
  </subkey>
</JAMSServers>
```

**Microsoft Messaging Queueing (MSMQ)**

JAMS uses Microsoft Message Queue (MSMQ) to reliably pass messages between the JAMS Services. It does this by creating a private queue named `JAMSRequests`.

MSMQ is included with Windows, but it is not installed by default. If MSMQ is not installed before installing JAMS, the JAMS installer will install MSMQ using the default minimum settings.

**Note:** if you want to control how MSMQ is installed it is best to install MSMQ before starting the JAMS installation.

If you want to change the way that MSMQ is installed after installing JAMS:

1. Stop the JAMS Services.
2. Reinstall MSMQ.
3. Then restart the JAMS Services.

**Configuration Settings**

JAMS contains a number of configuration settings. You can change these settings using the Configuration shortcut on the JAMS Menu.

The configuration settings are described in the table below:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contact</strong></td>
<td>Description</td>
</tr>
<tr>
<td>Company Name</td>
<td>The company name to include in error reports.</td>
</tr>
<tr>
<td>Contact Name</td>
<td>The person in your organization responsible for JAMS deployment.</td>
</tr>
<tr>
<td>Contact Email</td>
<td>The responsible person’s email address.</td>
</tr>
<tr>
<td>Contact Phone</td>
<td>The responsible person’s phone number.</td>
</tr>
<tr>
<td>Automatically Report</td>
<td>When set to true, failures of the JAMSServices will be automatically sent to JAMS technical support.</td>
</tr>
<tr>
<td>Exceptions</td>
<td><strong>E-Mail</strong></td>
</tr>
<tr>
<td>Description</td>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>

©2020 HelpSystems, LLC. All Rights Reserved.
Default Notify Email Address
A comma separated list of e-mail addresses that will receive all notifications. You can remove inherited addresses by prefixing the address with a minus (-) sign. You can remove all inherited addresses by entering -*.

From Address
The from email address used when sending the e-mail from JAMS.

Maximum Log Size
Log files that are larger than this value will not be attached to notification e-mails.

SMTPServer
The name of the SMTP server used for JAMS. This must be configured to send error reports to JAMS Support.

Configuration Options

Schedule Examples
Examples of natural language date specifications displayed in date editors.

Configuration ACL Security Bypass

Grant Administrators Bypass
Option to allow individuals in the Admin. group to bypass ACL security.

Grant Bypass Group
Individuals listed in a specified group to bypass ACL security.

Configuration Encoding

Default input Encoding
The default input encoding for routine jobs.

DefaultOutputEncoding
The default output encoding for routine jobs.

Configuration File Locations

Default Log Location*
The default location for Job log files.

Default Macro File
The default macro file used for parsing.

Temporary File Location
The location for temporary files.

Configuration History

History Lookback Period
The number of days in the past to query history inside a detail dialog.

History Query Limit
Maximum number of history records to return in a single query? (zero means unlimited)

History Query Time Limit
The maximum time to wait (in seconds) for a response to a history query.

Configuration Monitor

Keep Completed
Keep completed Jobs in the Monitor for a specified period of time.

Configuration Workdays

Work Monday
Is Monday usually a workday?

Work Tuesday
Is Tuesday usually a workday?

Work Wednesday
Is Wednesday usually a workday?

Work Thursday
Is Thursday usually a workday?

Work Friday
Is Friday usually a workday?

Work Saturday
Is Saturday usually a workday?

Work _Sunday
Is Sunday usually a workday?

Presentation

Description
**Theme**

- **Light Theme**: The theme shown to end-users when they select Light Theme.
- **Dark Theme**: The theme shown to end-users when they select the Dark Theme.

**Presentation**

**Documentation**

**Description**

- **Header**: This HTML fragment is placed before all documentation content configured within this server.
- **Footer**: This HTML fragment is placed after all documentation content configured within this server.

**Presentation E-Mail**

**Description**

- **E-Mail Header**: This HTML fragment is placed before all e-mail content configured within this server.
- **E-Mail Footer**: This HTML fragment is placed after all e-mail content configured within this server.

**Integrations**

**Paladin**

**Description**

- **Certificate**: The name of the x.509 certificate used to authenticate with Paladin.
- **Paladin User**: The Paladin User
- **Ask Job**: A Job that is submitted when we need to ask Paladin for a response.
- **Severity**: Only log files that are smaller than this maximum will be uploaded to Paladin.
- **Paladin URL**: The URL for the Paladin service

*The DefaultLogLocation is a .NET format string used to construct the full file specification for a Job's log file. The data values passed in the formatting operation are:

- 0. Directory specification
- 1. Filename
- 2. File Extension
- 3. Folder Name
- 4. Job or Sequence Name
- 5. Run occurrence number
- 6. JAMS Entry Number
- 7. Timestamp date & time
- 8. Timestamp year
- 9. Timestamp month
- 10. Timestamp day
- 11. Timestamp hour
- 12. Timestamp minute
- 13. Timestamp second
- 14. Timestamp millisecond

**Working with JAMS Services**

JAMS Scheduler installation includes four Windows Services:

- JAMS Scheduler Services
- JAMS Server Service
- JAMS Executor Service
- JAMS Agent Service

**Note**: Installing JAMS Agents on Windows also includes the JAMS Agent Windows Service.
JAMS Scheduler Service

The JAMS Scheduler Service is responsible for automatically scheduling jobs, firing triggers, and checking dependencies, etc. The JAMS Scheduler must include access to the JAMS Database or the service will fail. However, JAMS is designed to be resilient. Job execution is handled by the JAMS Executor service, so if the JAMS Scheduler service fails, no job execution information is lost.

JAMS Server Service

The JAMS Server provides middle-tier services to all JAMS Client components (GUI, Powershell, .NET Class Library, and Web services). While the JAMS Server service is not involved in the execution of Jobs, many jobs can use the JAMS PowerShell client, which does utilize this service.

JAMS Executor and Agent Services

The JAMS Executor is responsible for executing and monitoring Jobs. This service does not access the JAMS Database.

The JAMS Agent service is an extension of the JAMS Executor service. When the JAMS Executor needs to execute a Job on a different machine, it does so using the JAMS Agent running on that machine.

Troubleshooting JAMS Services

Each JAMS Service generates a ServiceName.log (e.g. JAMSScheduler.log) in the installation directory. These logs are reset every Sunday with the previous weeks log files renamed to ServiceNameArchive.log (e.g. JAMSSchedulerArchive.log).

Note: JAMS Services writes any serious errors to the Windows Event log. When troubleshooting JAMS, always check the event log and the .log files.

If you suspect there are problems with JAMS, shutting down one of the three services may resolve your issues. The list below provides some guidelines from the least to most disruptive option.

Restart the JAMS Scheduler

While the JAMS Scheduler Service does the most work, restarting it is the least disruptive. Restarting the JAMS Scheduler Service does not cause any Jobs failures and all job completion information remains intact. While the JAMS Scheduler service is stopped, no new Jobs can execute.

Stop the JAMS Server Service

The JAMS Server service may be stopped without losing any job execution information. The JAMS Client will not function while the JAMS Server service is shut down.

Shut down the JAMS Executor Service

Stopping the JAMS Executor service should be a last resort, as it will not typically resolve JAMS issues. This service executes and monitors Jobs. When this service is halted, completion information for any executing Jobs will be lost and some executing Jobs may fail. The JAMS Executor service does not access the JAMS database, so stopping this service won’t be necessary during SQL service maintenance.

Service Accounts

By default, JAMS services are set to run under the LocalSystem account, although this can be modified to run on a Windows Domain based account.

In general, it is recommended that you leave the JAMS Executor and JAMS Agent services running under LocalSystem. These services require access to the database or network and require the privileges associated with the LocalSystem account.

Use the Service Control application to change the account for the JAMS Scheduler and JAMS Server services in order to control network and database access.

When modifying the account, you may need to adjust the security settings on:
- C:\Program Files\MVPSI\JAMS\Scheduler folder
- C:\Program Files\MVPSI\JAMS\Scheduler\JAMSScheduler.log
- MSMQ JAMSRequests private queue
- JAMS Database

You will need to modify the security on the MSMQ JAMSRequests private queue in order to grant the domain account full access to the queue. This may require you to take ownership of the MSMQ queue.
The following local security policies should also be granted for the domain-based account:

- Log on as a Batch job
- Log on as a Service
- Adjust memory quotas for a process
- Bypass traverse checking
- Replace a process level token

If the domain based user account is not in the administrators group, create an Active Directory Group, than add the user to the group and make the following changes in the Common.config file located at: File\MVPSI\JAMS\Scheduler<directory>

<add key="AuthorizedGroup" value="domain\YourGroup"/>

Managing the JAMS Database

The JAMS Database provides critical features to the JAMS Scheduler and stores JAMS definitions. The following section contains information on managing the JAMS SQL database back end.

Manually Creating the JAMS Database

In most instances, the JAMS Database is created with the JAMS Scheduler during the installation process. However, in some cases, when installing the JAMS Scheduler you may prefer not to create the JAMS Database.

**NOTE:** The JAMS Database should always be created using the JAMS Installer or JAMS DBA utility. Manually creating the JAMS Database outside of these tools will cause various issues in JAMS.

JAMS looks for the SQL Server connection string in the Common.config file. This file is created during the database creation process, so if it is missing the JAMS Database does not exist. Likewise, if you want to recreate the database, simply delete or rename the file.

To create a JAMS database after the JAMS Scheduler has been installed, open a command window and execute the following commands:

```
CD "C:\Program Files\MVPSI\JAMS\Scheduler"
JAMSDBA INSTALL
```

During the installation process you will be required to provide SQL authentication information and use an account that has the ability to create a new database. To review these changes to the database, go to:

```
C:\ProgramFiles\MVPSI\JAMS\Scheduler\JAMS_DB_ERRORS.sql file
```

Security

JAMS supports either Windows Integrated Authentication or SQL Server Authentication. This can be modified after installation by changing the connection string found in the Common.config file. The default installation requires the JAMS Services to be running under the LocalSystem account. To facilitate this the installation executes the following SQL commands:

```
exec sp_grantlogin @loginame='BUILTIN\Administrators'
exec sp_grantdbaccess @loginame='BUILTIN\Administrators', @name_in_db='JAMSService'
exec sp_addrolemember @rolename='JAMSApp', @membername='JAMSService'
```

The net effect of using Windows Integrated authentication is that anyone in the administrators group can map the JAMSService Database user and become a part of the JAMSApp Database role.

**Note:** You can modify the security to fit your needs, but the JAMS Services must be included in the JAMSApp Database role.

Backing up the JAMS Database

It is critical to back up the JAMS Database. Since the JAMS Database is a standard SQL Server database, simply add it to your existing SQL Server backup procedures.

Choosing the Recovery Model

Before implementing a backup plan, you will need to choose which recovery model to use with the JAMS database.

The default installation uses the Simple Recovery Model. The other option is the Full Recovery Model. The main difference between the two is in the amount of journaling performed by the database.
Simple Recovery Model

The Simple Recovery Model is the easiest to use. You periodically backup the JAMS database only, not the journal file. The downside to this model is a lack of recovery options. If the JAMS Database becomes corrupted you can recover it from a backup but, all database changes since the last backup will be lost. This includes job execution history, so Job dependencies may not work as expected.

Full Recovery Model

This model contains more robust recovery features. However, the disadvantage of this model is that you must manage the journal file for the JAMS Database. Each time a change is made to the JAMS Database it is also written to the journal file.

For example, if a hardware failure occurs or you have otherwise corrupted the JAMS Database, you can restore a backup and then recover it using the journal files up to a specific point in time.

Restoring the JAMS Database from a Backup

Restoring a JAMS Database from a backup requires a standard SQL Server restore operation, but there are some details to consider before attempting the restoration process.

The JAMS Database includes the current schedule. Often, you may not want the current schedule to be restored. For example, if the database was backed up on Monday and you restore it on Friday, you probably don’t want to start running Monday’s Jobs along with the rest of the week’s processing.

CD C:\Program Files\MVPSI\JAMS\Scheduler
  JAMSDBA RESET SCHEDULE

CD C:\Program Files\MVPSI\JAMS\Scheduler
  JAMSDBA UPDATE/LOG

Moving the JAMS SQL Database

To relocate the SQL database used by JAMS:

1. Perform a normal SQL database backup and restore it to the target SQL server.
2. Edit the connection string property of the Common.config file found in: C:\Program Files\MVPSI\JAMS\Scheduler.
3. Restart the JAMS Scheduler service from the Windows control panel services applet.
4. Run JAMS_DB_ERRORS.sql on the new server. This script is found in the JAMS Scheduler directory.

Note: The process for moving a SQL/JAMS database does not require reinstalling JAMS.

The common.config file contains the connection string used by JAMS services to link to the SQL database. Below are two examples of connection strings; one for Windows Authentication models and another for SQL Authentication models.

Connection Strings for Windows Authentication

```xml
<appSettings>
  <add key="ConnectionString" value="Server=SQLA\INST1; Failover Partner=SQLB\INST1; Database=JAMS; Application Name=JAMS; Connect Timeout=600; Integrated Security=SSPI"/>
</appSettings>
```

Connection Strings for SQL Account Authentication

```xml
<appSettings>
  <add key="ConnectionString" value="Server=SQLA\INST1; Failover Partner=SQLB\INST1; Database=JAMS; Application Name=JAMS; Connect Timeout=600; Trusted_Connection=False;uid=YOURSQLACCOUNT;pwd=YOURPASSWORD" />
</appSettings>
```

SettingsPath for the JAMS Client

JAMS can override all saved client setting at any installed locations. This can be useful when you want multiple JAMS Clients to default to the same shortcuts and themes. It also simplifies how new layouts and shortcuts changes can be standardized across-the-network with only one location needed as a template.
The “SettingsPath” can be configured for each JAMS Client by modifying the user.config file as described below:

1. Open the configuration file which by default is located at: `C:\Program Files\MVPSI\JAMS\Client\User.config`

2. Add a new entry within the “appSettings” tags called “CommonSettingsPath” with a value of the location where you want to save the client settings. For example:

   ```xml
   <appSettings file="User.config">
   <add key="ClientSettingsProvider.ServiceUri" value="" />
   <add key="CommonSettingsPath" value="\\AppServer\ClientSettings" />
   </appSettings>
   
   add key="WriteToCommonSettings" value="true" />
   ```

3. Optionally, if you want the JAMS Client to save changes back to this common location simply add an entry for `WriteToCommonSettings` with a value of “true”. By default, any new changes will be saved to the user-specific location.

4. Copy the desired client settings into the new common location. By default, JAMS stores the client settings by version number for each user and stores it in the .XML files found in:

   `%HOMEPATH%\Local Settings\Application Data\MVP Systems, Inc\JAMS\X.X.X`.

   You can replicate settings from one client to the next by copying these flat files.

   ![File List](image.png)

   - Two critical files in this group include: `Servers.xml` and `JAMSPage.xml`. The former stores the JAMS server definitions and the latter defines shortcut settings on the JAMS page. In most cases, these are the only files that are required, with other files controlling more user specific setting such as columns and column widths.
   - Or you can copy the entire folder to the new location. These files contains the current client settings. But make sure to select the folder that corresponds to the current installed version of JAMS, e.g., “6.1.510.0”.
   - JAMS will attempt to load files from the common location that it does not find in the user-specific location. In order to force an existing JAMS client to load from the common location you will need to delete the user-specific settings.

   These user settings can be helpful if you want to have varying client layouts for different user groups such as administrators, developers, and users. For example, you could set up three directories that contain the appropriate settings for each group. After configuring the Client config file for the members of these groups, you could then manage the settings independently.

   - If the `WriteToCommonSettings` option is “true” for multiple JAMS Clients, they can overwrite the common settings.
   - JAMS will only load from the common location if it doesn’t find the files in the user-specific location.
   - You can prevent users from making changes to the server definitions by adding:

     ```xml
     <add Key="LockServer" value="true" />
     ```

   - Prevent users from changing shortcuts by adding:

     ```xml
     <add Key="LockShortcutbar" value="true" />
     ```

   - When Configuration files exist in the Common location and the user’s local settings, ensure the Client loads Configuration files from the Common location by adding:

     ```xml
     <add Key="PrioritizeCommonSettings" value="true" />
     ```

©2020 HelpSystems, LLC. All Rights Reserved.
The following is a sample of a Configuration file using PrioritizeCommonSettings:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
    <add key="PrioritizeCommonSettings" value="True" />
    <add key="CommonSettingsPath" value="\az-win019-test\share\ClientSettings" />  
    <add key="LockShortcutBar" value="False" />
    <add key="LockServer" value="False" />
    <add key="LockQuery" value="False" />
    <add key="LockUsername" value="False" />
    <add key="PromptForSearchKeys" value="False" />
    <add key="DefaultServer" value="" />
    <add key="DefaultPort" value="773" />
</appSettings>
```

**Managing Encryption Keys**

JAMS encrypts password and private key information when it is stored in the database. The standard JAMS installation uses a predefined encryption key, which is adequate for many sites. For additional protection you can generate a unique encryption key but must ensure that it is properly backed up and secured.

When generating a unique encryption key, JAMS uses Advanced Encryption Standard (AES) to re-encrypt all password and private key information within the database. The generated key is then encrypted and stored using the Windows Data Protection API (DPAPI). The protected key is then linked to user account associated with the JAMS Server and Scheduler services.

Use the following commands available in the JAMSDBA utility to manage encryption keys:

- **GENERATE KEY** - Generates a new 256-bit encryption key, decrypts the password with the old key then encrypts it with the new key. The new encryption key is then stored using DPAPI.
- **EXPORT KEY** - Pulls the encryption key from DPAPI and writes it to a text file.
  
  **Caution**: It is critical that you protect the EXPORT file since this key is not encrypted.

- **IMPORT KEY** - This is similar to the GENERATE KEY. In this case the new key is pulled from a text file instead of being generated. If recovering from backup or configuring the secondary server in a failover configuration, you should use the `/NOENCRYPT` qualifier to skip the decryption and re-encryption of the current data.

For additional information on protecting/restoring a JAMS Server as well as working in failover environments, go to the complete topic: JAMS Security: Managing Encryption Keys (on-line documentation).

**Protecting a Standalone JAMS Server**

To protect a standalone JAMS Server, follow the steps below:

1. Log on to the server as a user with "Execute" access to the JAMS configuration settings.
2. Set your default directory to the JAMS Scheduler installation directory. The default is: `C:\Program Files\MVPSI\JAMS\Scheduler`
3. Execute JAMSDBA.EXE to open a JAMSDBA> prompt.
4. Enter `GENERATE KEY` and press Return.
5. Enter `EXPORT KEY YourFileName.txt` and press Return.
7. Then move the exported text file to a safe location.

**Restoring a Standalone JAMS Server**
When restoring a backup of the JAMS database to different server, the encrypted passwords cannot be decrypted because of the encryption key DPAPI protection associated with the original machine/user. To restore the encryption key from a backup, follow these steps:

1. Log on to the server as a user with “Execute” access to JAMS configuration settings.
2. Set the default directory to the JAMS Scheduler installation directory. The default is: `C:\Program Files\MVPSI\JAMS\Scheduler`
3. Copy the exported text file (YourFileName.txt) from your safe location to the current directory.
4. Execute JAMSDBA.EXE to open a JAMSDBA> prompt.
5. Enter `IMPORT KEY/NOENCRYPT` and press Return.

### Failover Environments

To protect the servers in a failover environment please follow the steps listed below:

1. Make sure that your primary JAMS Server is in the running state.
2. Log on to the primary server as a user with “Execute” access to the JAMS configuration settings.
3. Execute JAMSDBA.EXE to open a JAMSDBA> prompt.
4. Enter `GENERATE KEY` and press Return.
5. Enter `EXPORT KEY` YourFileName.txt and press Return.
7. Move the exported text file to a safe location.
8. Log on to the secondary server as a user with “Execute” access to the JAMS configuration settings.
9. Set your default directory to the JAMS Scheduler installation directory. The default is: `C:\Program Files\MVPSI\JAMS\Scheduler`
10. Copy the exported text file (YourFileName.txt) from your safe location to the current directory.
11. Execute JAMSDBA.EXE to open a JAMSDBA> prompt.
12. Enter `IMPORT KEY/NOENCRYPT` and press Return.

### High Availability

It is always beneficial when a secondary Failover environment can be configured as part of your JAMS installation. The Failover provides a completely redundant instance of JAMS residing on a secondary server that relies on a heartbeat connection between the two. By default, this heartbeat is set to a 60 second interval. In the event the Failover does not get a response from the Primary server within 3 consecutive beats, the Failover will “takeover” as the Primary scheduler in order to maintain the integrity of the scheduler system.

### JAMS Failover Architecture

The JAMS Failover Architecture should consist of at least three servers:

- JAMS Primary Scheduler Server
- JAMS Failover Scheduler Server
- JAMS Agent Server(s), where Jobs are run

In the event of a failure, the Failover Scheduler will take over for the Primary Scheduler, ensuring the schedule of Jobs remains intact. If Jobs are executing locally on the Primary Scheduler Server, a failure would result in the failure of all Jobs executing on the Scheduler server. Running Jobs on JAMS Agent Server(s) insulates those Jobs from Primary Scheduler Server failure. To further insulate the executing JAMS Jobs from server failure, the JAMS Agent can be configured in a cluster.

### Installing and Configuring the JAMS Failover

The JAMS Failover Engine provides automatic failover for the JAMS Scheduler without using a Microsoft Cluster. Note that the JAMS Failover should **NOT** be configured in a cluster.

Follow the instructions below to install a JAMS Failover Server:

1. Begin by stepping through the normal installation of the JAMS Primary Engine.
2. For redundancy of the JAMS Database, you can configure AlwaysOn Availability (which replaced Database Mirroring starting with SQL Server 2012), after you complete these steps.
3. At this point, make sure to stop the JAMS Scheduler service on the Primary Engine machine.
4. Next, begin by installing the JAMS Scheduler on the second node. When prompted for the database server, specify the same SQL Server, Instance and Database Name as you did with the Primary JAMS Engine. After
choosing Next, a dialog will appear stating “Database already exists”. Be sure to select “Use Database.”

5. At this point, make sure to stop the JAMS Scheduler service on the secondary server, just as you did in the previous step with the Primary Server.

6. Create/Edit a Failover.config file (a sample is displayed below) in the \MVPSI\JAMS\Scheduler directory on the Primary server - by default the path is: C:\Program Files\MVPSI\JAMS\Scheduler\.

7. Copy the completed Failover.config from the Primary Engine to the Secondary Failover server. Make sure the Failover.config is identical on both servers.

8. Next, start the JAMS Scheduler server on both the Primary and Failover machines.

9. Lastly, if the Primary and Failover share a database, it will be necessary to add a user to the JAMS database to allow the Secondary engine to connect. This can be done by running the following SQL statements on the JAMS database.

```
Note: Make sure to replace the login name with your machine name as shown below:

exec sp_grantlogin @loginame='YourDomain\YourMachineName$
exec sp_grantdbaccess @loginame='YourDomain\YourMachineName$','', @name_in_db='JAMSMachine2'
exec sp_addrolemember @rolename='JAMSApp', @membername='JAMSMachine2'
```

A Sample Failover config

```
<FailoverConfig>
<Primary>Server1</Primary>
<Secondary>Server2</Secondary>
<Port>4773</Port>
<Interval>60</Interval>
</FailoverConfig>
```

Common Log Output Location

If you choose to have a common log output location on network share for both Primary and Secondary schedulers, please review the following article.

Setting a common log output location for JAMS in a Failover Environment

Clustering

JAMS Supports Windows Clustering. There are many ways to configure a Windows Cluster and your options can vary depending on whether you are configuring the JAMS Scheduler or the JAMS Agent.

JAMS Scheduler

The JAMS Scheduler can be configured in an active/passive mode. The JAMS Services should be included in the same cluster resource group to enable failover as a unit.

JAMS Agent using an Active/Passive Configuration

The JAMS Agent can be configured using an “active/active” or “active/passive mode”. The key is the IP address cluster resource. When directing Jobs or Queues to a clustered agent you can specify a DNS name that resolves to an IP address that fails over in a cluster to execute a Job on whichever node in the cluster is currently serving that IP address. This is considered an active/passive configuration.

JAMS Agent using an Active/Active Configuration

You could also specify a DNS name that resolves to an IP address that does not failover. In this way, you are directing the Job to a specific machine in the cluster. This is considered an active/active configuration.

In addition, you can create multiple IP address resources that normally runs on different nodes in the cluster, but fails over if a machine stops. This is also considered to be an active/active configuration.

SQL Server Mirroring

JAMS supports database mirroring. Please consult the SQL Server documentation for information on configuring and creating a mirrored database.

Once the JAMS Database has been successfully created, you should edit the connection string located in the Common.config file to add the "Failover Partner= OtherServer." This isn't strictly required, but if JAMS starts and
the primary database server isn't available, JAMS won't know where the secondary server is located.

Disaster Recovery Configuration

Recommended DR Model

The JAMS Server and Agents should have DR equivalents, named differently than the production servers. This will ensure the boxes are not seen on the network with the same name as their Production equivalents.

With the JAMS DR architecture set up as outlined above, the DR Process will consist of:

- Restoring the Production JAMS database to the DR Server and running any necessary database SQL scripts
- Replacing references to the JAMS Production nodes with DR equivalents

Disaster Recovery Process

Before beginning the DR Process

On the DR Server, ensure the following are true:

- The JAMS Services are disabled (Includes the JAMS Executor, JAMS Scheduler, and JAMS Server)
• Host files on the JAMS Server and Agent Nodes are pointing to the DR Servers
• Common.config is pointing to the local SQL Server (located at C:\Program Files\MVPSI\JAMS\Scheduler\Common.config by default)

1. Restore the SQL Database
First, create a backup of the JAMS Production Database, then restore the JAMS Production Database in DR.

2. Truncate the Failover Table
Truncate the failover table on the DR Server.

   NOTE: The failover table details the GUID of the JAMS server that is running against this database. This GUID should match the Installed GUID found in the Server Configuration file (located at C:\Program Files\MVPSI\JAMS\Scheduler\Server.config by default).

3. Backup the current Monitor View state
On the JAMS DR Server:
• Start the “JAMS Server” Service
• Within the JAMS DR Client, go to the Monitor View and take screenshots of the current Jobs within the Monitor View
• Stop the “JAMS Server” service

   NOTE: The Failover table on the DR Server must be truncated before the JAMS Server is started. The JAMS Server will then automatically add the required entries to the table.

4. Update the JAMS Server Agents/Nodes to point to the DR Servers
The JAMS Nodes must be updated to point to the DR Servers instead of Production Servers. In this example, all Sequences and Jobs within JAMS refer to a JAMS Queue, rather than point to an Agent Node directly. With this Queue configuration, re-directing the Nodes will be a simple case of updating the Production Queues to point to the DR Nodes. JAMS will not allow existing nodes to be deleted if any running or pending Jobs that point to those Queues/Nodes are within the Schedule/Monitor view. Therefore, in order to allow the nodes to be updated, all Jobs will need to be cleared from the Monitor View using the Reset Schedule process detailed below.

5. Reset the JAMS Schedule
On the JAMS DR Server
• Ensure the JAMS Scheduler Service is Stopped
• Run the JAMSDBA executable as an Administrator
• Type RESET SCHEDULE and hit Enter

   Note that when JAMS starts up against a restored database, it will check to see whether the Schedule is over 48 hours old. If it is, it will automatically rebuild the schedule from the current time, otherwise it will continue with the current schedule as-is. As the JAMS Database will likely be less than 48 hours old on DR, the ‘Reset Schedule” process will have to be manually run using the JAMS DBA utility.

   Once the Scheduler Service starts, the Schedule will be built based on the current time. In order to rebuild the Schedule against a different time, the /Restart switch can be specified.

6. Enable Access to the JAMS Client
Start the JAMS Server Service on the DR Server. In order to be able to successfully connect to the Server from a JAMS Client, the JAMS Server service will need to be enabled and started. The status of the JAMS Server can be seen at the bottom left hand corner of the JAMS client.

7. Update the JAMS Server Nodes
• Ensure the JAMS Client is opening against the DR Server and that all Jobs have been cleared from the Monitor View
• Run the “Update Queues” PowerShell script. This will update JAMS Queues to point to DR Nodes
NOTE: As the JAMS Scheduler will still be disabled, the Update Queues script must be run outside of JAMS.

- Ensure the Queues have been successfully re-pointed in the front end by viewing the Agents and Queue Nodes

8. Start the JAMS Services

On the DR Server, manually start the JAMS Scheduler Service and the JAMS Executor Service. Starting the JAMS Scheduler will result in the Schedule being rebuilt within the Monitor view, as defined by the JAMS DBA utility.

NOTE: Sequences or Jobs that were scheduled before DR that were still pending when production went offline may need manual attention.

9. Run Test Processes

On the DR Server, run a test process in JAMS to validate that the Scheduler is working as expected.

Licensing DR

The JAMS DR Server requires a unique JAMS License.

Agents will be consumed as follows:
The JAMS License is on the JAMS Server and it will detail the number of available Agent licenses. When the JAMS Server is restarted, the number of allocated Agent licenses is set to zero. As Jobs consume Agents, the allocated License count is incremented.

JAMS TCP/IP Ports

JAMS uses the following TCP/IP Ports for both outgoing and incoming connections.

JAMSAgent.exe: The JAMS Agent listens on port 77 for requests to execute Jobs from other machines running JAMS.
JAMSServer.exe: The JAMS Server listens on port 773 for connection requests from JAMS Clients.
JAMSScheduler.exe: The JAMS Scheduler listens on port 2773 for connections from the JAMS Server and JAMS Debugger. This is usually confined to the local machine.
JAMSExecutor.exe: The JAMS Executor listens on port 3773 for connections from the JAMS Scheduler and JAMS Debugger. This is usually confined to the local machine.
JAMSScheduler.exe: The JAMS Scheduler listens on port 4773 for connections from the Failover JAMS Scheduler. This is configured in the Failover.config file.

Common Configuration: If no port is specified in the common.config file in C:\Program Files\MVPSI\JAMS\Scheduler (default location), the default SQL port of 1433 is used.

Web Interface HTTP: The JAMS Web Interface uses port 80 by default for HTTP.
Web Interface HTTPS: The JAMS Web Interface uses port 443 by default for HTTPS.
Agents in JAMS

There are many different Agents available for JAMS, each designed and configured for a specific platform and purpose.

Most JAMS Agents are installed to the JAMS Server along with the JAMS Scheduler. The JAMS Client may be used to deploy and upgrade Agents, once those Agents have been defined.

Defining, Configuring, Deploying, and Deleting JAMS Agents

Agents in JAMS are initially defined using 4 fundamental properties:

- **Agent Name** - The reference name for the Agent definition in JAMS.
- **Agent Description** - An optional description of the defined JAMS Agent.
- **Agent Platform** - The operating system platform for the Agent. Options include Windows, Linux, UNIX, OpenVMS, and more.
- **Agent Type** - The type of Agent to be created, based on the Agent's intended use. The default Agent Type is Outgoing.

Define an Agent in JAMS

1. Select the Agents shortcut from the Menu.
2. In the Agent Definitions view, select Add from the Control Bar to open the Add an Agent Definition dialog.
3. In the dialog, give the new Agent a Name, select the Agent Type and Agent Platform, and if desired, a Description.
4. Click OK to save the initial Agent definition. By default, JAMS will automatically open the full Agent Definition dialog after the Agent is initially saved.

**NOTE:** The Name of a JAMS Agent is a friendly name for the Agent in JAMS. The JAMS name of an Agent is not required to match the system name of the Agent Machine.

Configure an Agent in JAMS

Once an Agent has been defined in JAMS, it can be configured using the Agent Definition dialog.

1. In the Agent Definitions view, select the Agent to be configured.
2. Select Properties from the Control Bar to open the Agent Definition dialog.
3. In the Properties tab, users will see a list of Agent Properties tailored to the Type and Platform of the Agent.
4. Use the available Agent Properties to configure the Agent.
   E.g. on an Outgoing Windows Agent, the DNS Address would be defined.
5. Select the Security Tab to modify group permissions to the Agent.
6. Select the State tab to view Agent information, or set a Job Limit for the Agent.
7. When configuration is complete, Save and Close the Agent Definition dialog.

Deploy an Agent in JAMS

Once an Agent has been defined in JAMS, it can be Deployed from the Agent Definitions view. The JAMS Primary Engine offers built-in capabilities to install a JAMS Agent on any machine running Windows, Linux, Solaris, AIX, System I (AS400), or a HPUX operating system.

If the Agent already exists, users may follow the steps below to upgrade the Agent.

1. In the Agent Definition view, select the Agent to be deployed.
2. Select Deploy from the Control bar to open the Deploy JAMS Agent dialog.
3. In the dialog, select the JAMS User that will be used when deploying the JAMS Agent. This user should have all necessary permissions to the defined JAMS Agent.
4. Click OK.

**NOTE:** When initially deploying an Agent to a Linux/Unix machine, the JAMSAgentX Job will fail on the
monitor the first time it executes. This is because JAMS initially caches the host fingerprint. Upon redeployment, JAMS Agents can also be manually installed. For more information, please use the following links:

- [JAMS Agent for Windows Manual Installation](#)
- [Manual Installation of JAMS Agent for Unix and Linux](#)

**NOTE:** OpenVMS Agents cannot be push deployed from the JAMS Server.

**NOTE:** for more information on working with specific JAMS Agents, please refer to the Agent topics.

### Delete an Agent in JAMS

Any JAMS Agent can be Deleted from the Agent Definitions view. Note that removing the Agent from the JAMS GUI will not remove the JAMS Agent software from the given Agent Machine.

1. In the *Agent Definition* view, select the Agent to be deleted.
2. Select *Delete* from the Control bar, or *Right-Click* the desired Job, then select *Delete* from the *Agent Options* menu to open the *Delete Agent* dialog.
3. If the desired Agent was selected for deletion, click *Yes*. If the incorrect Agent was selected, click *No* or *Cancel*.

### Agents and Queue Definitions (Delete an Agent from a Queue)

JAMS Jobs can be submitted to run on Queues with specified Agent nodes. To remove an Agent from a Queue, open the Queue Definition View and double-click the desired Queue. On the Queue's definition *Status* tab, click on the box with the black arrow pointing to the agent node listing, then press the *Delete* key to remove it from the list. Select either the *Save* or *Save and Close* buttons at the top of the definitions window to complete the process.

### Defining Incoming Agents

1. Navigate to C:/Program Files/MVPsi/JAMS/Scheduler/ on the JAMS Server and run JAMSRegister.exe
2. Enter the following code in JAMSRegister to create a root certificate authority, if one does not already exist.
   ```
   create root ca /storekey /makedefault
   ```
3. Add an incoming Agent executor with the following in the JAMS Register:
   ```
   add exec incoming /newcert /fw
   ```
4. On the Agent machine, install the agent software, then open JAMSRegister.exe from C:/Program Files/MVPsi/JAMS/Agent
5. In JAMS Register on the Agent machine, enter:
   ```
   add agent incoming /reqcert/server="SERVERPATH" /prompt /agentname="AGENTNAME"
   ```
   Where SERVERPATH is the url for the rest interface, and AGENTNAME is the name for the new incoming agent.
6. When prompted for a password, enter the password used to log onto the JAMS Server.
7. On the JAMS Server, check the agent list and ensure the new Agent appears on the Agent list, with a state of *Online*.  

JAMS Agent for OpenVMS

The JAMS Agent for OpenVMS is an optional component that provides background processes for executing JAMS Jobs under the direction of a JAMS Scheduler located on a different machine running Windows or OpenVMS.

Prerequisite Software

This version of the JAMS Agent for OpenVMS requires:

- OpenVMS/VAX version 6.2 or higher
- OpenVMS/AXP V6.2 or higher or . . .
- OpenVMS/Itanium V8.2 or higher

Decompressing Zip Files

The JAMS Agent comes as a self-extracting zip archive when downloaded over the Internet. Once decompressed the file is named: JAMSAGENT??042x.EXE where "??" is "AXP," "IA," or "VAX" and 'x' is replaced by the patch level of the kit.

These self-extracting zip archives are OpenVMS executable images and are unpacked by running them. The archives contain the following files:

- JAMSAGENT??042.A
- JAMSAGENT??042.B

License Registration

The JAMS Agent can be installed and run on any machine. All license restrictions occur on the machine running the JAMS Scheduler. The JAMS Agent does not require a LMF License PAK.

Installation Procedure Requirements

The installation takes 5-10 minutes depending on the type of media and your system configuration. Before installing JAMS, you must have the following privileges and resources:

- SETPRV privileges or CMKRNL, WORLD, and SYSPRV privileges.
- At least 1000 free blocks of disk space on the device to hold the JAMS executable files.
- A minimum of 50 free global pages.
- A minimum of 100 free global sections.

The installation procedure initially checks for the required resources before the installation. However, if any resources are deficient, the procedure will issue an error message and identify the resource(s) before terminating the installation.

To determine the number of free global pages on your system, type the following DCL command:

$ WRITE SYSSOUTPUT F$GETSYI("FREE_GBLPAGES")[Return]

The number displayed equals the total number of free global pages. If this number is too small, you can increase the GBLPAGES system parameter by editing the file SYSSYSTEM:MODPARAMS.DAT and invoking the SYSSUPDATE:AUTOGEN.COM command procedure.

For more information on adjusting SYSGEN parameters and using AUTOGEN, refer to the appropriate VMS documentation.

To determine the number of free global sections on your system, type the following DCL command:

$ WRITE SYSSOUTPUT F$GETSYI("FREE_GBLSECTS")[Return]

The number displayed equals the total number of free global sections. If this number is too small, you can increase the GBLSECTIONS system parameter by editing the file SYSSYSTEM:MODPARAMS.DAT and invoking the SYSSUPDATE:AUTOGEN.COM command procedure.

For more information on adjusting SYSGEN parameters and using AUTOGEN, please refer to the appropriate VMS documentation.
Using VMSINSTALL

VMSINSTAL is a command procedure used to install software onto an OpenVMS operating system. In step 2 of the installation process, described below, you must invoke VMSINSTAL. The format of the VMSINSTAL command is as follows:

$ @SYS$UPDATE:VMSINSTAL upivvv ddnn:[dir] [OPTIONS N][Return]

upi: is the unique product identifier; in this case, JAMSAGENTAXP, JAMSAGENTIA, or JAMSAGENTVAX.

vvv: is the placeholder for the version number of the product. Refer to the label on the JAMS distribution media to determine the version you will be installing. For example, the complete product name for Alpha JAMS Agent Version 4.2 is: JAMSAGENTAXP042.

ddnn:[dir]: represents the device and directory location for the distribution kits. If installing from a CD-ROM, the distribution kits are located in the [JAMS042.KIT] directory. If JAMS was downloaded over the Internet, insert the device and directory where you unpacked the Zip archives. DKA400:[JAMSAGENT042.KIT] is the device and directory used in the example on the following page.

Options N

If this optional parameter is supplied, the installation procedure will ask to display or print the release notes before performing the actual installation.

When invoking VMSINSTAL, the installer will check to see if:

- You are logged into the system manager's account. You should install layered software from the system manager's account on the default device with the directory set to SYSSUPDATE.
- You have adequate quotas for installing layered products.
- There are any user processes running on the system.

If VMSINSTAL detects any user processes running on the system, you will be asked if you want to proceed with the installation.

To continue, type YES and press Return. If you choose to stop the installation, press Return.

The JAMS Agent Installation Process on OpenVMS

To install the JAMS Agent on OpenVMS, follow the steps below.

To abort the installation procedure at any time, press CTRL/Y. The installation procedure cleans up any files it has created and returns you to the DCL prompt.

1. Log in to a privileged account and set your default device and directory to SYSSUPDATE.

2. Invoke VMSINSTAL

   $ @VMSINSTAL JAMSAGENTAXP042 DKA400:[JAMS040.KIT] [Return]

3. Answer the standard VMSINSTAL questions. VMSINSTAL will ask if you are satisfied with the backup of your system disk. Also, it may ask additional questions if you are not logged on as SYSTEM or it detects other unusual conditions.

4. Provide a device name for the JAMS executables. You will be asked what disk device should be used for the JAMS executables.

   Note: You can install Alpha, Itanium, and VAX executables on the same device. JAMS will create different directories for each installation.

5. Provide a device name for JAMS data. You will then be requested to select a disk device to use for the JAMS data files.

   Note: You can store the data files on the same disk as the executables. However, the JAMS data disk must be the same on all nodes in the VMScluster.

6. Start JAMS when the installation is complete? To execute the JAMS startup procedure when the installation
is complete, press Return; otherwise enter No and press Return.

Note: To abort the installation procedure at any time, press CTRL/Y. The installation procedure cleans up any files it has created and returns to the DCL prompt.

7. Purge Files? Do you want to purge the files replaced by this installation. To purge, press Return; otherwise, enter No and press Return.

8. Run the IVP? Do you want to run the installation verification procedure (IVP) after the installation. To proceed, press the Return key; otherwise, enter No and press Return.

9. Read the informational messages. As the installation proceeds, scan the on-screen informational messages for potential problems. JAMS puts all files in its own directory tree with the exception of the following files:

    SYS$STARTUP:JAMS_AGENT_STARTUP.COM
    SYS$STARTUP:JAMS_SITE_STARTUP.COM

Configuring and Managing the JAMS Agent for OpenVMS

After successfully installing the JAMS Agent, it is important that you perform the following tasks.

Insert JAMS_AGENT_STARTUP.COM into System Start-Up

You must execute the JAMS Agent startup procedure on every node that will be using the JAMS Agent. If you install the JAMS Agent on a node that also has the JAMS Agent already installed, execute the JAMS_AGENT_STARTUP.COM procedure before starting any batch queues.

To ensure that the JAMS Agent startup procedure is executed every time the system reboots, insert the following line into the system startup procedure: SYS$MANAGER:SYSTARTUP_VMS.COM

$ @SYS$STARTUP:JAMS_AGENT_STARTUP.COM

Note: Make sure to insert the above line after the disk containing the JAMS data files has been mounted and before initiating any batch queues.

The JAMS startup procedure can also be defined as a startup file using the VMS SYSMAN utility.

Initialize the JAMS Agent on all Nodes

The JAMS Agent is fully initialized on the node where the installation is performed. In order to initialize the JAMS Agent on other nodes in a VMS cluster, issue the following command for every node where the JAMS Agent will execute:

$ @SYS$STARTUP:JAMS_AGENT_STARTUP.COM

Check TCP/IP port 77

The JAMS Agent listens on TCP/IP port 77. Make sure this port isn’t blocked by a firewall. You can modify the port setting by defining a logical name, for example:

$ DEFINE/SYSTEM/EXECUTIVE JAMS_AGENT_PORT 7777

In this example, the command would tell the JAMS Agent to listen on port 7777. Make sure to restart the JAMS Agent after defining the logical name.

Managing the JAMS Agent

You can run JAMS_EXE:JAMS_AGENT_MANAGER_EXE to manage the JAMS Agent. The utility has commands for starting, stopping, and displaying the status of the JAMS Agent. You can get more information by running the utility and using the Help command at the JAMS_AGENT> prompt.

Error Conditions
If the installation procedure fails for any reason, a message is displayed and the installation will terminate. Try to determine what caused the installation to fail. If you can correct the problem, do so and then restart the installation.

If you cannot determine the cause of the problem or have any questions about JAMS, please contact JAMS technical support at:

**Phone:** (800) 261-5267  
**E-Mail:** Support@JAMSScheduler.com

---

**NOTE:** When configuring the OpenVMS Agent on the JAMS Scheduler, users must type **OpenVMS** into the Agent **Platform** field.
JAMS Agent for System i

The JAMS Agent for System i is designed to support the IBM System i operating system (aka i5/OS and OS/400). JAMS connects to the IBM System iServer using TCP/IP protocols.

Deploying the JAMS Agent for System i

The JAMS Agent for IBM System i is installed along with the JAMS Scheduler, but must be pushed out to the agent machines. Specifically, deployment is controlled from the Agents view in the JAMS Client.

Before attempting to deploy, make sure you have provided each JAMS user the appropriate System i access along with the privileges required to deploy the agent.

NOTE: If the library JAMSLIB is present in both the system library list and user library list, remove JAMSLIB from the system library list.

Upgrading the JAMS Agent for System i

Installing an updated version of JAMS may already include the JAMS Agent. To deploy the update, select one or more agents from the Agent Definitions view, right-click, and select Deploy.

Troubleshooting the JAMS Agent for System i Deployment

The most common issues with the JAMS Agent deployment are:

- The user selected for the deployment job does not have sufficient privileges on the target machine.
- FTP settings are disabled or blocked.

To gain a more detailed view of the deployment, review the Job Log of the DeploySystemi in Job History.

If the JAMS DeploySeriesi job does not work because of access restrictions (e.g., the FTP is blocked), you can manually deploy the agent by following steps described below:

1. Create a JAMS user profile: CRTUSRPRF USRPRF(JAMS) PASSWORD(*NONE) SPCAUT(*JOBCTL)
2. Create a library: CRTLIB LIB(JAMSLIB)
3. Create a save file: CRTSAVF FILE(JAMSLIB/JAMSSAVF)
4. Copy the JAMSSAVF.SAVF from the JAMS Scheduler to the AS/400 server (JAMSSAVF.SAVF is located at C:\Program Files\MVPSI\JAMS\Scheduler\Agents\IBMSeriesi by default)
5. Restore the objects: RSTOBJ OBJ(*ALL) SAVLIB(JAMSLIB) DEV(*SAVF) SAVF(JAMSLIB/JAMSSAVF) MBROPT(*ALL) RSTLIB(JAMSLIB)
6. Start the JAMS subsystem: STRSBS SBSD(JAMSLIB/JAMS)
JAMS Agent for Windows

The JAMS Agent for Windows includes background services that execute JAMS Jobs under the direction of a JAMS Scheduler on a different machine.

**Manually Installing the JAMS Agent for Windows**

1. To manually install the JAMS Windows agent, go to the JAMS Primary Engine (Scheduler) machine and navigate to the following path: `Program Files\MVPSI\JAMS\Scheduler\Agents\Windows`
2. Once there, obtain either the `SetupAgent.msi` or `SetupAgentx64.msi` file and copy it to the remote Agent machine.
3. From the remote agent machine, double-click to run the installer and follow the prompts. When the installation is complete you will see a Windows service called 'JAMS Agent' running.
4. The installer package set up an agent service on the machine called 'JAMS Agent', and a directory is expanded containing the executable files in `C:\program files\MVPSI\JAMS\Agent` by default.
5. The JAMS Agent service listens on port 77 in order to communicate with the JAMS Scheduler.
6. The agent must then be defined in the JAMS Client.
   - Select the Agents shortcut and click the Add button on the Ribbon bar to open the Agent Definition Wizard.
   - Enter the appropriate information to define the agent.

   **Note:** the agent deployment is not necessary, leave the Deploy JAMS Agent box unchecked, as you have already performed a manual install of the agent.

7. It is recommended that the JAMS Agent Service run as 'Local System'.

**Push Deploy the Windows Agent using Local User Accounts**

When deploying a remote JAMS Agent to a Windows server, the standard practice is to use a domain account user with Administrator privileges on both the JAMS Primary Engine (Scheduler) and the remote JAMS Agent machine.

However, if you don’t have an environment that uses domain accounts, there are two alternative options for setting up the remote JAMS Agent software for a Windows server:

- Manually install the JAMS Agent for Windows (see the preceding section) or . . .
- Modify the registry setting changes on the Agent Machine. In order to allow the push deployment of the Windows JAMS Agent to a remote server when not using domain user accounts, make sure that the JAMS Primary Engine (Scheduler) and remote Agent server utilize the same administration level user account and password. Once that step is complete, perform the following registry edit on the remote agent machine:

  **CAUTION:** Improperly modifying the registry can harm your system. It is always recommended that you back up your registry prior to making any changes.

1. Browse to the following folder: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\system`
2. Right-click a blank area in the right pane.
3. Click New.
4. Click DWORD Value and Name it `LocalAccountTokenFilterPolicy`.
5. Double-click the item you just created and Type “1” into the Value Data box and click the OK button.
6. Restart your computer.

You should now be able to push deploy the JAMS Agent to the remote machine.
JAMS Agentd for Linux/Unix

Introduction

Agents allow users to define specific actions that can run locally or remotely throughout a JAMS network. By default, JAMS can only run Jobs and Sequences on the primary engine. Agents allow users to extend that capability to run Jobs on a variety of different environments. For example, by deploying JAMS Agents, a single JAMS Scheduler can execute tasks on many systems.

The JAMS Agentd is a version of the JAMS Agent which runs as a background process on Linux/Unix machines. This topic details how to install this unique Agent to your primary machine.

NOTE: Jobs running on an AgentD Agent must use either the UnixShell or UnixScript execution methods.

Installing the Agent

1. To get started, select the appropriate jams-agentd package (either the .rpm [Red Hat packages] or .deb [Debian packages]) from the JAMS download site at: http://download.jamsscheduler.com/repo/ and install it on your Linux system as root. This will create the /etc/jams.d/jams-agentd.config set up file using the default settings and should also automatically start the daemon. JAMS AgentD must be installed as root. To run JAMS AgentD as a user, refer to the Running JAMS AgentD as a Non-Root User section on this page.

Both Debian and Red Hat repositories are available for packages by MVP Systems Software, Inc.

These repositories use 0x0744742A as signing key.

Repositories

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debian</td>
<td>Debian Repo for use with Debian, Ubuntu, Mint, and others.</td>
</tr>
<tr>
<td>Red Hat</td>
<td>RPM Repo for use with Red Hat Enterprise Linux, Fedora, CentOS, Oracle Linux, and others.</td>
</tr>
</tbody>
</table>

Debian Packages

Packages

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Description</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>jams-agentd</td>
<td>Linux Agent Daemon for JAMS Scheduler</td>
<td>amd64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i386</td>
</tr>
<tr>
<td>jamsagentx</td>
<td>Linux On-Demand Agent for JAMS Scheduler</td>
<td>amd64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i386</td>
</tr>
<tr>
<td>jamsagentx-static</td>
<td>Linux On-Demand Agent for JAMS Scheduler, built statically</td>
<td>amd64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i386</td>
</tr>
</tbody>
</table>

Red Hat Packages
The jams-agentd daemon can be started and stopped using the `chkconfig` command on Red Hat-derived systems and with the `invoke-rc.d` command on Debian-derived machines.

**Note**: Making any modifications to the config file will require a daemon restart. The config file includes comments describing the various available options.

### Running JAMS AgentD as a Non-Root User

Normally, jams-agentd will run as root in order to allow it to authenticate and run Jobs as other users. However, it can also be run as a non-root user by disabling the root services for jams-agentd.

In this mode, all Jobs will be run as the executing user, irrespective of the requested user name for that Job. The config file and all working directories are then located in the jams directory on the user's home directory.

To run jams-agentd as a non-root user, the following steps must be performed:

1. Install JAMS AgentD as root using the installation instructions above
2. Stop and disable the root service as root
   
   **Disable Run as Root for AgentD**
   ```
   jams-agentd --kill-daemon
   jams-agentd --remove-service
   ```

3. Become the user that will run jams-agentd
4. Run the following commands:

   **Run jams-agentd as non-root**
   ```
   # create a directory for JAMS
   mkdir ~/jams
   # create the specified conf file based on the user
   jams-agentd -C ~/jams/jams-agentd.config
   # read and modify jams-agentd.config for your environment
   # Check the listening port, as non-root users cannot listen on ports less than 1024
   # Start jams-agentd daemon
   edit the .conf file
   jams-agentd -d
   # jams-agentd will create all other directories it needs to run as a non-root user
   ```

5. To stop running jams-agentd as the user:

   **Stop the Installer**
   ```
   jams-agentd -K
   ```

#### Packages

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Description</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSAgentX</td>
<td>Linux On-Demand Agent for JAMS Scheduler</td>
<td>i386, x86_64</td>
</tr>
<tr>
<td>JAMSAgentX-static</td>
<td>Linux On-Demand Agent for JAMS Scheduler, built statically</td>
<td>i386, x86_64</td>
</tr>
<tr>
<td>jams-agentd</td>
<td>Linux Agent Daemon for JAMS Scheduler</td>
<td>i386, x86_64</td>
</tr>
</tbody>
</table>
6. Ensure that jams-agentd is started every time the system restarts.

NOTE: It may be necessary to use the full pathname for the daemon when executed by a non-root user.

Public Key Authentication

PUBLIC-KEY is a valid user_authentication option in the config file.

PAM Support

Currently, the PAM support needs to have a PAM configuration file for jams-agentd. This is expected to be in the /etc/pam.d directory. When installing jams-agentd, the installer will attempt to create this file by copying the file for sshd with certain edits.

The PAM support only uses auth and account parts. No password updates are possible, nor is a session used.

It should be possible to authenticate with SSH using PAM (with the pam_ssh.so module), but this is not the same as the PUBLIC-KEY authentication. In this case, the JAMS User would need to have a Password that unlocks the Private Key on the target machine, but not a Key Management setting.

Sample Code

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c/-config-file</td>
<td>Use an alternate configuration file</td>
</tr>
<tr>
<td>&lt;config_file&gt;</td>
<td></td>
</tr>
<tr>
<td>-d/-daemon-mode</td>
<td>Fork/exit the program to run in the background</td>
</tr>
<tr>
<td>-l/-log-level &lt;level&gt;</td>
<td>Set the logging level to ERROR, WARN, INFO, DEBUG, or TRACE</td>
</tr>
<tr>
<td>-p/-port-number &lt;port&gt;</td>
<td>Listen on alternate port</td>
</tr>
</tbody>
</table>

Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c/-config-file</td>
<td>Create config_file with default settings or send to standard out</td>
</tr>
<tr>
<td>&lt;config_file/&gt;</td>
<td></td>
</tr>
<tr>
<td>-i/-install-service</td>
<td>Install jams-agentd as an auto-started service</td>
</tr>
<tr>
<td>-K/-kill-daemon</td>
<td>Kill running daemon</td>
</tr>
<tr>
<td>-P/-pam-file &lt;pam_file&gt;</td>
<td>Create pam configuration file for jams-agentd or send to standard out</td>
</tr>
<tr>
<td>-Q/-query-status</td>
<td>Test if the daemon is currently executing</td>
</tr>
<tr>
<td>-R/-remove-service</td>
<td>Remove the auto-started jams-agentd service</td>
</tr>
<tr>
<td>-S/-service-file</td>
<td>Create service start-up script or send to standard out</td>
</tr>
<tr>
<td>&lt;service_file/&gt;</td>
<td></td>
</tr>
<tr>
<td>-U/-update-config</td>
<td>Update an existing config_file with new configuration settings</td>
</tr>
<tr>
<td>&lt;config_file&gt;</td>
<td></td>
</tr>
<tr>
<td>-V/-version</td>
<td>Print the current version</td>
</tr>
</tbody>
</table>
z/OS Integration Pack

JAMS for z/OS incorporates z/OS Job scheduling into the JAMS platform and requires nothing to install on the z/OS host. When creating the agent used with the z/OS integration, you must select the z/OS agent type and the FTP platform.

Use this step-by-step guide to define a new z/OS Job using the JAMS Client.

Setting up the Connection with JAMS as a TSO User

To set up a connection with JAMS you will need to first define a TSO user on the mainframe as well as in JAMS (from the Credentials shortcut).

1. Open the Credential Definitions view from the JAMS Menu.
2. Click the Add button on the Control Bar.
3. The JAMS Credential definition dialog opens.
4. Set the Credential Name, Logon As, Password, and Password Confirmation values.
5. Click OK. By default, JAMS will open the full Credential Definition dialog after the Credential definition is initially saved.
6. On the Security tab of the Credential dialog, Add the ACE for the TSO user.
   Please note: this user must have login rights as well as rights to execute items that will be added in later JAMS jobs.
7. Save and Close the Credential definition.


JAMS interfaces with JES as a TSO user. The TSO login ID is interpreted by JAMS as a valid JAMS User, which is encrypted on the JAMS side and controlled by RACF on the z/OS side. The TSO user must have permissions to submit and receive Jobs on both platforms and the TSO username must contain 7 characters or less.

On the JAMS side, the z/OS Integration must be installed on the JAMS Server.

Setting up a z/OS Job in JAMS

1. To create a z/OS Job in JAMS
2. Select the desired folder for the z/OS Job, then ensure the Job Definitions tab is selected.
3. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
4. In the dialog, give the new Job a Name, Description (optional), and Execution Method.
   In this case, select z/OS to create a z/OS Job.
5. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
6. The Job Definition dialog will open.
   NOTE: An Execute As property should be configured with the previously defined TSO user.
   NOTE: An Agent property should be configured with the Mainframe Agent hostname or IP address.
8. On the Source tab, insert the JCL code.
9. Save and Close the Job definition.

How does JAMS run a Mainframe Job?

Jobs are stored on an FTP server located on the mainframe using a proprietary IBM-based FTP language. JAMS communicates with the mainframe using this FTP language. Once a Job is executed in JAMS an FTP connection is opened to the FTP Daemon in JES and the Job is sent to the mainframe and executed.

JAMS monitors the Job spool output and waits for the Job to complete by polling the FTP connection. Once the output files are complete JAMS purges the files from the JES spool.

Steps can be run in JES in the same way as tasks in a JAMS Sequence. JCL defines the behavior of the Job depending on what is happening with a particular Job task.

What Information does JES Return?

The JAMS Scheduler retrieves all Job related output from the JES Spool. This includes any DD statements with SYSOUT held on the spool. The output appears in the JAMS log and can be used by operations administrators to
debug problems and, if necessary, restart the Job.

JAMS for z/OS parses the Job output and provides job success ("0") or failure information for the JAMS Scheduler to react and report on a variety of message codes, including HASP and return code analysis.

What happens when a Job fails?

If a Job fails due to a network outage or other problem all relevant information is included in the output and also displayed in the JAMS log. Customization of these messages can be set using a configuration file.

A special option in the Job definition includes a listing of Jobs on the JES Spool allowing a Job to rerun on the JAMS side for output recovery.

JAMS for z/OS Capabilities

Manages and Submits JCL Files to JES2 on z/OS for Execution

JAMS provides the ability to store a JCL source in its internal database. The JCL then becomes a part of the JAMS Job Definition along with other parameters and information needed to run the job.

Schedules Jobs with Dependencies on Multiple z/OS Hosts

The JAMS Scheduler can support multiple z/OS hosts and track Job dependencies between them and non-z/OS hosts.

The JAMS Scheduler Retrieves and Displays the Job Output

The JAMS Scheduler retrieves all job related output from the JES Spool. This includes any DD statements with SYSOUT held on the spool. The output appears in the JAMS log and can be used by operations administrators to debug problems and, if necessary, restart the jobs.

Includes Automatic Determination for all Job Results

JAMS for z/OS parses the job output and provides job success or failure information for the JAMS Scheduler to react and report on a variety of message codes, including HASP and return code analysis.

Utilizes Standard JES Settings

JAMS for z/OS uses JESINTERFACELEVEL 1 and doesn’t require special customization of Parmlib settings.

Requires No installation on z/OS Hosts

JAMS for z/OS uses z/OS FTP with JES and requires no additional mainframe software installation.

Provides Automatic Job Purging Capabilities

JAMS for z/OS automatically purges the job output files for the JES2 Spool after each Job has completed and been retrieved.

Provides Reliable Tracking of Jobs Running on the Mainframe

The JAMS Scheduler can recover its connection to JES if it is lost due to a network outage or other problem, even when a job is in the middle of executing on z/OS. In the event of failure on the JAMS server, JAMS can restart for that specific job and seamlessly recover.

Uses a Secure TSO User Login

JAMS interfaces with JES as a TSO user. The TSO login ID is interpreted by JAMS as a valid JAMS user and encrypted on the JAMS side and controlled by RACF on the z/OS side.

Provides Full Integration with the JAMS Scheduler

JAMS for z/OS becomes part of the integrated capabilities of the JAMS Scheduler across different platforms allowing it to control jobs based on dependencies on all supported hosts.
JAMS for z/OS is Easy to Set Up

Setting up a z/OS job is now as easy as defining any other JAMS Job.

Provides a Listing of Pending Jobs on the z/OS Spool

A special option in the JOB definition includes a listing of JOBS on the JES Spool allowing a job to rerun on the JAMS side for output recovery.

Scenario: Synchronizing Data with JAMS for z/OS

Suppose an organization has the need to reliably synchronize data from an OLTP system based on a MS SQL Server or Oracle with a mainframe-based database. This scenario is easily implemented with JAMS for z/OS.

In order to accomplish this the JAMS Scheduler would initially run a job, based on an event, that generates files comprised of data extracted from a MS SQL Server database. It would then upload those files to the z/OS platform via FTP before submitting JCL through JAMS for z/OS. These uploaded files would then update a mainframe database. The reverse scenario is just as easy where a MS SQL Server is updated with files generated by the mainframe.
Integration Packs
Banner in JAMS

Introduction

This integration allows JAMS users to take advantage of Banner's core functionality such as registration, grade distribution, human resource information, financial aid processing, and other financial information.

The JAMS integration acts as an intermediary between JAMS and Banner. When the adapter runs, it initiates communication with the Banner application server and instructs executing Banner jobs as configured in JAMS. The process can then be monitored, canceled, and restarted in either Banner or JAMS.

Setting up Banner to work with JAMS

1. **Install AgentX on the Banner application server.** This agent can be automatically deployed or manually installed, specifically where the gjajobs.shl is stored.

   Use the links at the end of this topic to learn more about deploying JAMS Agents.

   ![NOTE: If the Banner server is running Windows, users may deploy a JAMS Windows Agent to the Banner server.]

2. **From the JAMS Client, select the Credentials Shortcut and set up a JAMS Credential that has permission to access the Banner application server and can also connect to the remote Linux server. This can also be the same credential that was used to deploy the agent.**

3. **Configure the Banner Execution Method in JAMS.** From the JAMS Client select the Execution Methods Shortcut to access the list of available Execution Method options.

4. **Double-click the Banner Execution Method from the list to open its properties and select the Template tab.**

5. **Customize the template settings in the areas shown in the following screenshots.**
   
   Line 6 through 16 - Set the source options for Oracle and Java.
   
   Line 21 - Source the Banner environment.
   
   Line 26 - Change the value to the Banner directory.
   
   Line 28 through 33 - This section serves to increment the Banner One-Up-Number. This section should not need to be changed.
   
   Line 35 through 40 - This section ensures the parameters on the JAMS Job are passed to the Banner server. This section should not need to be changed.
   
   Line 44 - Ensure gjajobs.shl is referenced properly. Users may need to SSH into the Banner server and manually confirm this information.
   
   Line 53 through 66 - This section pulls the .lis and .log files from Banner. This section should not need to be changed.
NOTE: The Banner Environment may also be set using a bash profile (`source ~/.bash_profile`) if the
Banner environment variables have been set on the default bash profile.

**NOTE:** The Banner directory as well as the directory containing the gjajobs.shl need to be configured.

**NOTE:** If sensitive data would be included in the Banner .log and .lis files, users may wish to create a copy of the Banner Execution Method, then remove the logging options from the copy’s template. This new Banner Execution Method could be used in situations where Banner writes sensitive data to the Banner output files.

6. Make sure that any folder containing a JAMS Banner Job (including sub-folders) includes a path to the Banner Macro. This path must be populated in the Folders Properties (Properties -> Source Options -> Template Library).

By default, this path is `C:\Program Files\MVPSI\JAMS\Scheduler\BannerMacros.xml`

7. With the configuration complete, create a new JAMS Job and chose the Banner Execution Method in the Job Definition dialog.

8. On the Job definition’s source page, set the Banner User, select a Job Type from the list, and then enter the Script Name.

   **Note:** The script name should not include the folder path or the .shl extension. The script name in JAMS must be in the same case as what is stored on the Banner server.

9. Double-click the newly completed JAMS Job to access its properties. Select the Properties tab and navigate to the Submit Options section. Edit the Job by selecting the Execute As and Agent properties. The Execute As should be a UNIX account with access to JAMS AgentX.

**Banner Parameters**

Banner parameters always have an associated number, as seen in the left side of the example below.

To pass parameters from JAMS to Banner, simply create JAMS Parameters with names that match the Banner parameter numbers on the given Job. Set the default value of the JAMS Parameter to what the value would be in Banner. Finally, users may set the Parameter Prompt in JAMS to match the Banner parameter name.

This is how the above Banner parameters would look when set on a JAMS Job:
Users may also specify multiple values for a single parameter by separating the values with pipe characters "|". For instance, if parameter 01 had a value of |Val1|Val2|Val3|, JAMS would send the following to Banner:

01 = Val1
01 = Val2
01 = Val3

Parameter 88 is special as it can include a "label" before each value. JAMS pulls off the value and then inserts it to the 88 parameter so a value of “This,That” will be passed as: 88This = That.

For example, a value of |One,Federal|Two,Grant is passed as:

88One = Federal
88Two = Grant

You can also define the parameter as 88This and then the value only needs to contain “That”.

©2020 HelpSystems, LLC. All Rights Reserved.
Dynamics AX Integration Pack

Configuring the .NET Business Connector

To ensure JAMS will run Dynamics AX Jobs, a Local Client (Where JAMS will retrieve the Job List) and a Business Connector (Where JAMS will run Jobs) must be properly configured.

Begin by opening Microsoft Dynamics

- Open the Start Menu
- Navigate to Administrative Tools
- Open Microsoft Dynamics AX Configuration

NOTE: To properly configure Dynamics AX with JAMS, ensure Dynamics AX and the installed JAMS Add-on are for the same MS Dynamics AX version. Installers for Dynamics AX versions can be found on the JAMS Support site.

Configure the Local Client

- Set the Configuration Target to "Local Client" using the drop-down menu
- In the Connection tab, verify the Application Object Server and Instance Name are correct.

If the server name cannot be changed, a new configuration must be created.

- Click the Manage button, then select Create Configuration
- In the dialog box, give the configuration a name, select Active Configuration, then click OK
- Highlight the Server Name in the list of available servers and click Edit.
• In the dialog box that appears, fill in the Server Name, Instance Name, TCP Port and WSDL Port, then click OK.

• Click Apply
• Click Refresh Configuration
• Click OK to save the Configuration

NOTE: You will need the TCP/IP and WSDL ports open from the machine running the JAMS client to the AOS server in order to create Dynamics AX jobs.

Configure the Business Connector

The Business Connector Configuration Target is the AOS where the JAMS job will connect to when the job actually runs.

• Set the Configuration target to “Business Connector” using the drop down menu.
• In the Connection tab, verify the Application Object Server is correct and that the Instance Name is correct.

If the server name cannot be changed, a new configuration must be created.

• Click the Manage button, then select Create configuration
• Give the configuration a name, select Active Configuration, then click OK
- Highlight the **Server Name** in the list of available servers and click **Edit**.
- In the dialog box that appears, fill in the **Server Name**, **Instance Name**, **TCP Port** and **WSDL Port**, then click **OK**.
Creating a User Credential for Dynamics AX Job Execution

JAMS natively supports automation using many leading business applications, including Microsoft Dynamics AX solutions. Integration Packs for these and other preferred applications are selected during the JAMS installation process. Once the integration is enabled, users can create, manage, deploy and monitor almost any kind of Dynamics AX job.

The JAMS Integration Pack for Microsoft Dynamics AX allows users to initiate Dynamics AX processes and take advantage of monitoring and report distribution capabilities using JAMS scheduling software.

When the integration runs, it initiates communication with and instructs Dynamics AX to schedule, execute AX jobs, batch jobs and reports as configured in JAMS. The process can then be monitored, canceled and restarted in either Dynamics AX or JAMS.

Follow the steps below to quickly add new JAMS user credentials.

1. Open the Credential Definitions view from the JAMS Menu.
2. Click the Add button on the Control Bar.
3. The JAMS Credential definition dialog opens.
4. Set the Credential Name, Logon As, Password, and Password Confirmation values.

**NOTE:** The user defined here must have the authority to use the .NET Business Connector.

A Microsoft Dynamics AX User is only allowed to use the .NET Business Connector if they satisfy one of the following Criteria:
5. Click **OK**. By default, JAMS will open the full **Credential Definition** dialog after the Credential definition is initially saved.

6. On the **Security** tab of the Credential dialog, **Add the ACE** for any user that is already set up in AX. Please note: this user must have login rights as well as rights to execute items that will be added in later JAMS jobs. This may require setting up additional users based on their access rights in the AX product.

7. **Save and Close** the Credential definition.

### Setting up the AX Job

Users must define and enter specific attributes in order to successfully run a job in JAMS.

Once a job has been properly defined it can either:

1. Be submitted to run on demand or . . .
2. Set up to run automatically using a schedule.

**Note:** A single JAMS Job can execute many other jobs. A schedule can exist on different machines to allow for separate management tasks and eliminate single points of failure.

### Creating a New Folder

The first step is to create a new folder or alternatively use an existing folder directory to locate similar JAMS jobs.

**Note:** Creating new or using existing folders in JAMS is optional.

To create a new JAMS definitions folder follow the steps below:

1. Open the **Definitions** view from the JAMS Menu.
2. **Right-Click** the desired Parent Folder in the Folder Navigator, then select **Add Folder** from the drop-down list.
3. The **Add a new Folder** dialog will appear.
4. Give the new folder a **Name**, then click **OK** to save the Folder definition.
5. Once the folder is defined, right-click on the folder and select **Properties** from the drop-down list to access the full **Folder Definition** dialog.
6. Define additional Elements, Parameters, Security, and Properties on the folder as desired, then click **Save and Close**.

### Adding a new Dynamics AX Job using the JAMS Definition Wizard

After creating a folder, you can quickly set up a Dynamics AX job using the **Add a New JAMS Job Definition** dialog.

Follow the steps below to create and define a JAMS/AX job:

1. Select the desired folder for the Dynamics AX Job, then ensure the **Job Definitions** tab is selected.
2. Click the **Add** button from the Control Bar to open the **Add a New JAMS Job Definition** dialog.
3. In the dialog, give the new Job a **Name**, **Description** (optional), and **Execution Method**.
   - In this case, select **MSDAX2012Job** to create a Microsoft Dynamics AX Job.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click **OK**.
5. The **Job Definition** dialog will open.
   - **NOTE:** Ensure “Execute As” has been set to the Dynamics AX user Credential.
7. On the **Source** tab, enter the appropriate values for the given Dynamics Job.
   - **Company:** Select the AX company ID for this job.
   - **Job Type:** Choose the appropriate radio button.
   - **Job:** Choose from a list of defined jobs in the AX AOT
   - **Report:** Select from a list of reports that are on an AX menu. This type of job can have many defined parameters.
   - **Refer to the topic:** Report Job Type Specific Parameters.
   - **Batch Job:** Choose from the list of batch Jobs (by name) that have been created in AX. A batch job contains all the pertinent information to execute on its own.
8. **Save and Close** the job definition.
Available AX Job Parameters

Generic to all Job Types

Include MS Dynamics Log

Debug

Language: Specify the language setting to be used when logging into AX Job.

ObjectServer: Specify the object server setting to be used when logging into AX Job.

Configuration: Specify the object server setting to be used when logging into AX Job.

Report Job Type Specific Parameters

Query: Specify the name of a query to be applied to the report before rendering it. This Query is set up in AX.

Destination: Specify where the report should be sent once rendered.

Possible Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Printer</td>
</tr>
<tr>
<td>3</td>
<td>Email</td>
</tr>
<tr>
<td>4</td>
<td>File</td>
</tr>
</tbody>
</table>

Printer  Printer
Email  Email
File  File

Default value:
1. Printer

FileFormat: Specify the format of the file being exported by processing a report.

Possible Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>Ascii</td>
</tr>
<tr>
<td>RTF</td>
<td>Rich Text Format</td>
</tr>
<tr>
<td>HTML</td>
<td>HTML</td>
</tr>
<tr>
<td>PDF</td>
<td>PDF</td>
</tr>
<tr>
<td>PDFEMBEDFONTS</td>
<td>PDF with embedded fonts</td>
</tr>
<tr>
<td>TEXTUTF8</td>
<td>Text in UTF8 encoding</td>
</tr>
<tr>
<td>1</td>
<td>Ascii</td>
</tr>
<tr>
<td>2</td>
<td>Rich Text Format</td>
</tr>
<tr>
<td>3</td>
<td>HTML</td>
</tr>
<tr>
<td>4</td>
<td>PDF</td>
</tr>
<tr>
<td>5</td>
<td>PDF with embedded fonts</td>
</tr>
<tr>
<td>6</td>
<td>Text in UTF8 encoding</td>
</tr>
</tbody>
</table>

Default value:
1. Ascii

FileName: Specify the name of the file the report will be exported to when the destination is set to File.

PrinterName: Specify the printer name report will be printed when the destination is set to Printer.

FirstPage:
- Specify
- Default
Subject: Specify the subject of an Email. Used when the destination is set to Email.

To: Enter an email address to send the report,

**Job Specific Parameters**

Parm: Specify a single parameter to be sent to a Job. Note the job in AX must be designed in such a way to accept a 'parm'
Automating a JD Edwards Job in JAMS

There are three prerequisite steps that must be undertaken before users can automate a JD Edwards job in JAMS.

- You must first install the JDE Integration Pack on the JAMS Scheduler and on the remote machine where runubexml.exe resides.
- Also, the JAMS Agent must also be installed on the remote machine where runubexml.exe resides.
- On the root installation directory (C:\Program Files\MVPSI\JAMS\Scheduler) of your JAMS installation, as well as the root installation folder of the JAMS agent (C:\Program Files\MVPSI\JAMS\Agent) locate the file named JAMSJDEHost.dll.config.sample and ensure the proper connection string for the JDE database has been defined. The following example utilizes a Microsoft SQL instance:

  ```xml
  <add key="ExeHost" value="MyJDEAlias"/>
  ```

  Note: If the JD Edwards system is configured with an alias for the JDE Batch Server in the application, a key must be added to the JAMSJDEHost.dll.config file.

  Additionally, if running JDE with an Oracle backend users can include an Oracle-based connection string as shown in the example below.

  ```xml
  <add name="jdeSystem" connectionString="Server=(local); Database=JDEDB; Integrated Security=SSPI"
  providerName="System.Data.SqlClient"/>
  ```

  Once the requirements for the connection string have been satisfied, make sure to rename the file upon saving to JAMSJDEHost.dll.config to allow JAMS to pick up and read the configuration settings.

  The binPath value must be set which is the location of runubexml.exe, as shown in the example below: Example: E:\JDEdwards\E910\system\bin32

  The Schema value must be set based on the given schema for the Master Control Table. E.g. if the Master Control Table's Schema is SVM920.F986110, set the Schema value in JAMSJDEHost.dll.config to be SVM920.

  ```xml
  <add key="Schema" value="SVM920"/>
  ```

  Note: This file must be adjusted on the Remote Agent side if JD Edwards jobs are running on a JAMS Remote Agent.

For the next step, create a new job definition and set the Execution Method as JDEJob. The job's source will display a fill-in-the-blank form requiring various inputs necessary to automate the JDE jobs.

Finally, add Home Directory and Execute As properties to the Job. Modify the Job's Properties to define a Home Directory other than the default System32. Utilizing a directory such as C:\JAMS_TEMP\ as well as defining an Execute
AS User Account that has permissions to run this job on Windows is recommended (See Defining RunAs User Accounts in JAMS).

This is the user accessing the C:\JAMS_TEMP directory on the remote JDE batch server, as well as accessing the Master Control Database from within JDE if the Connection String is set for Integrated Security.
Informatica Cloud In JAMS

The Informatica Cloud Execution Method allows users to automate any existing Informatica Cloud task, utilizing Informatica’s REST API.

In This Topic:

- Installing the Informatica Cloud Add-on
- Configuring the Informatica Cloud Execution Method
- Creating Informatica Cloud Jobs in JAMS

Installing the Informatica Cloud Add-on

1. Run the JAMS Installer
2. On the Feature Selection step, ensure the JAMS Add-on for Informatica Cloud component is checked
3. Complete the installation

NOTE: Un Checking components will uninstall those components.

Configuring the Informatica Cloud Execution Method

1. Run the JAMS Client as an administrator
2. Select the Execution Methods shortcut from the JAMS Shortcuts Menu.
3. Double click the InformaticaCloud execution method to open the Execution Method dialog.
4. Select the Parameters tab, then double click the InformaticaEndpoint parameter to open the Parameters dialog.
5. Set the Default Value to the InformaticaEndpoint value that reflects the given environment
6. **Save and Close** the *Parameters* dialog
7. **Save and Close** the *Execution Method* dialog

**NOTE:** The user running Informatica Cloud Jobs must have full access to the Informatica Cloud.

### Creating an Informatica Cloud Job in JAMS

1. Select the desired folder for the Informatica Cloud Job, then ensure the Job Definitions tab is selected.
2. Click the **Add** button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a Name, Description (optional), and Execution Method. In this case, select InformaticaCloud to create an Informatica Cloud Job.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
5. The Job Definition dialog will open.
7. On the Source tab, set the User (must have access to Informatica Cloud environment) and Task Type for the relevant Informatica Cloud Task.
8. The **Task** list will automatically populate with all of the Tasks of the selected **Type**. Select the relevant **Task** using the drop-down list.
9. When the relevant Informatica Cloud task has been configured in the new JAMS Job, click **Save and Close**
Micro Focus in JAMS

In this Topic:
- Installing the Micro Focus Integration Pack
- Configuring Micro Focus Execution Method
- Creating Micro Focus Jobs

Installing the Micro Focus Add-on

To work with Micro Focus in JAMS, users must install the JAMS Add-on for MicroFocus on the JAMS Scheduler Server, and must install a JAMS Agent and the JAMS Add-on for MicroFocus on the Micro Focus Server. If the JAMS Scheduler and Micro Focus are installed on the same server, an Agent is not required.

Install the Micro Focus Add-on on the JAMS Scheduler Server

1. On the JAMS Scheduler Server, run the JAMS Installer.
2. On the Feature Selection step, ensure the JAMS Add-on for MicroFocus component is checked in addition to the required components. Do not un-check any components unless those components should be uninstalled.
3. Finish the installation.

**NOTE:** Un Checking components in the Feature Selection step will uninstall those components.

**NOTE:** The JAMS Server and Micro Focus Server should be running the same version of JAMS.

Install the JAMS Add-on for MicroFocus on the Micro Focus Server

1. On the Micro Focus Server, run the JAMS Installer.
2. On the Feature Selection step, ensure the JAMS Agent and JAMS Add-on for MicroFocus components are checked.
3. Finish the installation.

Deploy a JAMS Agent to the Micro Focus Server

1. Start the JAMS Client as an Administrator.
2. Select the Agents shortcut from the Menu.
3. In the Agent Definitions view, select Add from the Control Bar to open the Add an Agent Definition dialog.
4. In the dialog, give the new Agent a Name, select the Agent Type and Agent Platform, and if desired, a Description.
5. Click OK to save the initial Agent definition. By default, JAMS will automatically open the full Agent Definition dialog after the Agent is initially saved.
6. Set Agent Properties in the Agent Definition dialog, then Save and Close the Agent Definition dialog.
7. In the Agent Definition view, select the Agent to be deployed.
8. Select Deploy from the Control bar to open the Deploy JAMS Agent dialog.
9. In the dialog, select the JAMS User that will be used when deploying the JAMS Agent. This user should have all necessary permissions to the defined Micro Focus Agent.
10. Click OK.

Configuring the Micro Focus Execution Method

1. Run the JAMS Client as an Administrator.
2. Select the Execution Methods shortcut from the Shortcuts Menu.
3. Double-click the MFJob Execution Method to open the Properties dialog, then select the Parameters Tab.
4. Use the Properties button to open the Parameter Properties dialog. Configure Micro Focus parameters as needed for the given Micro Focus environment.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFBINPort</td>
<td>Enter the MFBINP listener port. The default in JAMS is 9122. Listener configurations are stored in the Micro Focus listener configuration file, seelistener.exe.config. If a listener port must be created for JAMS, reference the Micro Focus documentation.</td>
</tr>
<tr>
<td>MFServer</td>
<td>Enter the name of the Micro Focus Server.</td>
</tr>
<tr>
<td>MFSubmitType</td>
<td>Set the relevant Submit type from JES2 (Job Entry Subsystem 2), JES3, or VSE. The default is JES2.</td>
</tr>
</tbody>
</table>

5. Save and Close the Parameters dialog, then Save and Close the MFJob Execution Method.

Creating a Micro Focus Job

1. Select the desired folder for the MicroFocus Job, then ensure the Job Definitions tab is selected.
2. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a Name, Description (optional), and Execution Method. In this case, select MFJob to create a Micro Focus Job.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
5. The Job Definition dialog will open.
   NOTE: An Execute As property should be configured with a user that has access to the Micro Focus Server.
7. Paste or write the desired code in the Source.
8. **Save and Close** the Job definition.

**NOTE:** The JCL Code in JAMS should match the syntax of Micro Focus code.

For more information on Micro Focus, consult the [Official Micro Focus Documentation](#).
The JAMS Management Pack for MS SCOM

The JAMS Management Pack provides monitoring of JAMS components and services using Microsoft SCOM 2007R2, 2012 and 2012 R2. The monitoring includes availability, health and performance data collection. In addition to these monitoring capabilities, the Management Pack supports Agent tasks and Dashboard views. The Management Pack supports monitoring of JAMS on the Windows operating system only. Support for other operating systems may be added at a later date.

Preparing to Import the Management Pack

To ensure that target JAMS computers are managed by the SCOM Agents (preferably using LocalSystem account), the account used by the SCOM Agent must have adequate permissions to access JAMS program directory, the system registry and access JAMS data via its PowerShell provider.

Note: You can use local accounts on managed computers and/or dedicated domain accounts to execute Management Pack rules and monitors; however, you will also need to configure the JAMS ‘Run As’ profile account using SCOM’s operations console as outlined below.

It is recommended the target computer with the JAMS Scheduler role have the PowerShell feature enabled or installed. Please refer to the JAMS Installation Guide for additional information. It is also recommended that you install the JAMS Instrumentation Integration for each managed JAMS Scheduler machine in order to reduce overhead and improve functionality of the Management Pack.

Importing the Management Pack

The Management Pack consists of a single file: MVP.JAMS.mp


1. Log on to the computer using an account that is a member of the Operations Manager Administrators role within the Operations Manager management group.
2. From the Operations console, click Administration. When you run the Operations console on a computer that is not a management server, the 'Connect To Server' dialog box appears. In the Server name text box type the name of the management server that you want the Operations console to connect to.
3. Right-click the Management Packs node and then click Import Management Packs.
4. The Import Management Packs wizard opens. Click the Add button and then select Add from disk.
5. The Select Management Packs to import dialog box appears. Change to the directory that contains the JAMS Management Pack file. Select MVP.JAMS and then click Open.
6. On the Select Management Packs page the management pack that you selected for import should be listed. Choose Import.
7. The Import Management Packs page appears and displays the import progress. If there is a problem at any stage of the import process select the management pack to view the status details. Click Close.

Discovery

After importing the JAMS Management Pack, SCOM runs a discovery process on all managed Windows computers in order to locate the JAMS installations. By default SCOM discovers all JAMS installations on Windows. No additional configuration should be necessary for most customer environments.

Customizing the Management Pack

The JAMS Management Pack is ‘sealed’, preventing any setting changes in the original Management Pack file. However, you can still perform and save customizations such as overrides for monitors, alert rules and views, and save them to a different management pack.

By default SCOM saves all customizations to the Default Management Pack. As a best practice you should instead create a new management pack for each sealed management pack you want to customize.

For more information about management pack customizations and the Default Management Pack, link to the article: [http://go.microsoft.com/fwlink/?LinkId=217006](http://go.microsoft.com/fwlink/?LinkId=217006)

The Management Pack includes the following customizations:

- JAMS Scheduler Performance Monitor. Note that there are 2 monitors targeting the JAMS Scheduler with and without the Instrumentation Integration installed (JAMS Instrumented Scheduler and JAMS Non-Instrumented Scheduler). Both monitors have identical thresholds for the number of currently executed or pending Job with a specific status.

You can disable these monitors and/or provide the overrides for the frequency as well as warning and error
thresholds. The screenshot below illustrates the Scheduler Performance Monitor parameters that can be overridden using the SCOM Operations Console.

- There are 2 alert rules that are defined for the Instrumented Scheduler class:
  - JAMS Scheduler Execution Warning
  - JAMS Scheduler Execution Error
These rules can be enabled or overridden to produce Scheduler alerts based on the level of events logged to MVPSI JAMS/Operational by the JAMS Instrumentation Integrations.

- JAMS Scheduler (performance) collection rules can be disabled or the frequency (IntervalSeconds) can be adjusted.

Note: there are separate set of rules for Instrumented and Non-Instrumented Scheduler classes.

- New monitors and rules can be easily added using new performance counters and/or ETW events implemented in the JAMS Instrumentation Integrations.

Note: corresponding providers are available only on Scheduler machines that have Instrumentation Integrations installed. These JAMS Scheduler instances are discovered as instances of ‘JAMS Instrumented Scheduler’ class derived from ‘JAMS Scheduler’ class.

The JAMS Management Pack creates ‘Run As’ profile named ‘Account for JAMS Collection Rules’ as shown on the
You can associate one or more SCOM accounts with this profile to define local or domain accounts that should be used by the management pack rules and monitors on each computer. Refer to SCOM documentation for further information on ‘Run As’ configuration.

Removing the Management Pack

For additional information on removing the Management Pack, go to the article: http://technet.microsoft.com/en-us/library/cc974489.aspx

Configuring a JAMS Scheduler to Send Alerts

1. In the MVPSI\JAMS\Scheduler directory, rename the Event.SCOM.default to be Event.SCOM.config.
2. Add an Alert Element on a JAMS Folder or Job that defines the text that you want to send in the Alert. To apply the alert to all folders, add the Alert element to the root folder.

JAMS looks for the following parameters before sending an alert to SCOM. These parameters can originate from the Job’s execution method, the Job itself, or the folder the Job is located in.

- **SCOMPriority**: sets the priority of the alert. This parameter should be one of three integer values, with 0 = low, 1 = medium, and 2 = high.
- **SCOMSeverity**: sets the severity of the alert. The parameter also should include one of three values, with 0 = informational, 1 = warning, and 2 = critical.
- **SCOMCustomField0** through **SCOMCustomField9**: these fields are passed to SCOM and intended for end-user customizations. The values are strings with JAMS performing symbol substitution, allowing you to reference any JAMS property using the standard <<JAMS.Name>> syntax.

View Information in the Operations Manager Console
Once imported, the JAMS Management Pack creates a root folder, **MVP JAMS**, that can be accessed via the SCOM Operations Console Monitoring pane using the following subfolders and views:

![JAMS Management Pack](image)

These Folders and Views display different aspects of monitoring data.

- **Root**: MVP JAMS folder
- **Alerts**: Alert View for all JAMS Alerts
- **Computer**: State view for all computers within the JAMS computer group.
- **Tasks**: Task view of all JAMS Scheduler and Agent tasks queued via the SCOM Operations console to run on SCOM Agent computer.
- **Components folder**
  - Scheduler State View
  - Agent State View
  - Client State View
- **Health Monitoring Folder**
  - Agent Health dashboard view displays JAMS Agent State View and Alerts.
  - Scheduler Health dashboard view displays JAMS Scheduler state views and alerts.
- **Performance Folder**
  - Scheduler Performance View

### JAMS Management Pack Agent Tasks

The JAMS Management Pack provides a set of admin tasks that can be queued to run on a SCOM Agent machine; for example, to start/stop the JAMS component services, query or submit JAMS Jobs, etc.

Some of these tasks are available via the Management Pack Alerts Knowledge links, while you can access JAMS Agent or Scheduler tasks via the Operations console Actions pane in the Management Pack Agents and Schedulers status views.

The following tasks are available for **JAMS Agents**:

- Start JAMS Agent Service
- Stop JAMS Agent Service

The following task are available for the **JAMS Scheduler**:

- Get Scheduler Jobs
- Restart Jobs
- Resume Jobs
- Start JAMS Scheduler Executor Service
- Start JAMS Scheduler Server Service
- Start JAMS Scheduler Service
- Start Job
Troubleshooting the Management Pack

If the Scheduler Performance Monitor and/or collection and alert rules are not working it may indicate that PowerShell is not available or JAMS PowerShell module is not installed on the target system. Please check the SCOM Active Alerts view and the SCOM application event log on the management server or management computers for the alerts or logged event notifications indicating PowerShell is not properly installed on the target computer or script execution errors such as insufficient Windows or JAMS permissions assigned to accounts used to run the management pack workflows.

By default these workflows use the account assigned to the SCOM Agent (such as LocalSystem), but you can customize the ‘Run As’ settings as described in the topic: Importing and Customizing the JAMS Management Pack for MS SCOM.

Troubleshooting Alerts

If you are not receiving alerts from JAMS to SCOM please check the following:

- Review the four steps made in the first section of this topic: Configuring a JAMS Scheduler to Send Alerts.
- Check the security permissions on the jamstoscom msmq private queue. Make sure that your SCOM activity user or Run As user has full control over the queue.
- If you don't have a Run As account configured for JAMS in SCOM, the default Activity user(s) needs to have full control over the jamstoscom queue.
NeoBatch In JAMS

In This Topic:
- Installing the NeoBatch Integration Pack
- Configuring NeoBatch Execution Methods
- Creating NeoBatch Jobs

Installing the NeoBatch Integration Pack
To work with NeoBatch in JAMS, complete the following installations, then deploy an Agent to the NeoBatch Server.

On the NeoBatch Server:
- NeoBatch Add-on
- JAMS Agent

NOTE: The NeoBatch Server must be running NeoBatch Server Edition. Developer or Client editions of NeoBatch will not function with JAMS.

On the JAMS Clients that need to edit NeoBatch Jobs:
- NeoBatch Add-on
- JAMS Client

NOTE: If the JAMS Scheduler is installed on the NeoBatch Server, an Agent is not required.

Install the NeoBatch Integration Pack and JAMS Agent on the NeoBatch Server
- Run the JAMS Installer.
- On the Feature Selection step, check the JAMS Agent and JAMS Add-on for NeoBatch components.

NOTE: Un Checking components will uninstall those components.

NOTE: The JAMS Server and NeoBatch Server should be running the same version of JAMS.

Install the JAMS Add-on for NeoBatch on JAMS Clients
- Run the JAMS Installer.
- On the Feature Selection step, check the JAMS Client and JAMS Add-on for NeoBatch components.

NOTE: Un Checking components will uninstall those components.

Deploy a JAMS Agent to the NeoBatch Server
1. Start the JAMS Client as an Administrator.
2. Select the Agents shortcut from the Menu.
3. In the Agent Definitions view, select Add from the Control Bar to open the Add an Agent Definition dialog.
4. In the dialog, give the new Agent a Name, select the Agent Type and Agent Platform, and if desired, a Description.
5. Click OK to save the initial Agent definition. By default, JAMS will automatically open the full Agent Definition dialog after the Agent is initially saved.
6. Set Agent Properties in the Agent Definition dialog, then Save and Close the Agent Definition dialog.
7. In the Agent Definition view, select the Agent to be deployed.
8. Select Deploy from the Control bar to open the Deploy JAMS Agent dialog.
9. In the dialog, select the JAMS User that will be used when deploying the JAMS Agent.
This user should have all necessary permissions to the defined NeoBatch Agent.
10. Click OK.

### Configuring NeoBatch Execution Methods

Before creating Jobs with NeoBatch in JAMS, users must configure the NeoBatch Execution Methods. This will allow JAMS to connect to the NeoBatch Database when running Jobs.

1. Run the JAMS Client as an administrator.
2. Select the **Execution Methods** shortcut from the **JAMS Menu**.
3. Double-click the **NeoBatchCatalog** execution method to open the **Execution Method Definition** dialog.
4. Select the **Parameters** tab, then double click on any parameter to open the **Parameter Definition** dialog.
5. Configure the NeoBatch parameters as needed for the given NeoBatch environment:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeoBatchClassName</td>
<td>Any alphanumeric string as determined by NeoBatch classes - the name of the mainframe class name that NeoBatch Jobs will run as.</td>
</tr>
<tr>
<td>NeoBatchDbConnectionString</td>
<td>The connection string to the NeoBatch Database. Update this for the given environment.</td>
</tr>
<tr>
<td>NeoBatchSubmitAssembly</td>
<td>The assembly used by NeoBatch. e.g. (Alchemy.NeoBatch.Submit, Version=3.2.1.30, Culture=neutral, PublicKeyToken=722c951c23d7c1ea)</td>
</tr>
</tbody>
</table>

The NeoBatch Submit Assembly Version and Key Token can be found by running the following powershell script. Note that the PowerShell script should be edited to reflect the path to Alchemy.NeoBatch.Submit.dll for the given environment.

```
Return Submit Assembly and Public Key Token
([system.reflection.assembly]::loadfile("c:\MyDLL.dll")).FullName
```

6. Click the **Save and Close** button to finalize the changes.
7. Repeat Steps 3-7 for the **NeoBatchJcl** execution method.

### Creating NeoBatch Jobs in JAMS

With the Execution Methods properly configured in JAMS, NeoBatch Jobs may be run by:

- Using **NeoBatchJcl** to run JCL code directly in JAMS
- Using **NeoBatchCatalog** to reference and run an existing Job in the NeoBatch catalog

### Run JCL directly in JAMS

1. Select the desired folder for the NeoBatch Job, then ensure the Job Definitions tab is selected.
2. Click the **Add** button from the Control Bar to open the **Add a New JAMS Job Definition** dialog.
3. In the dialog, give the new Job a **Name**, **Description** (optional), and **Execution Method**. In this case, select **NeoBatchJcl** to create a NeoBatch JCL Job.
4. By default, the full **Job Definition** dialog will open when the Job is initially saved. Click **OK**.
5. The **Job Definition** dialog will open.
   **NOTE:** An **Execute As** property should be configured with a user that has access to the NeoBatch Server.
7. Paste or write the desired JCL code in the **Source**.
8. Save and Close the Job definition.

NOTE: Hosting JCL in JAMS allows users to use the JAMS Audit Trail to monitor JCL code changes.

Run a NeoBatch Job from the NeoBatch Catalog

1. Select the desired folder for the NeoBatch Job, then ensure the Job Definitions tab is selected.
2. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a Name, Description (optional), and Execution Method. In this case, select NeoBatchCatalog to create a NeoBatch Catalog Job.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
5. The Job Definition dialog will open.
6. Define Elements, Parameters, Security, Properties, and Documentation as desired. NOTE: An Execute As property should be configured with a user that has access to the NeoBatch Server.
7. In the Job Source, set the path to a desired NeoBatch Job Member in the NeoBatch Catalog. This Job will be run when the JAMS Job Runs.
8. Save and Close the Job definition.
Netezza in JAMS

In This Topic:
- Installing the Netezza Integration Pack
- Creating Netezza Jobs

Installing the Netezza Integration Pack
To use the JAMS Add-on for Netezza, users must complete the following installations:
- Install the JAMS Scheduler and JAMS Add-on for Netezza on the JAMS Server
- Install JAMS AgentX on the Netezza Server, or on a Linux Machine with access to the Netezza Server

Installing Netezza for JAMS
1. On the JAMS Scheduler Server, run the JAMS Installer.
2. On the Feature Selection step, ensure the JAMS Integration Pack for Netezza Integration Pack is checked. Do not un-check any components unless those components should be uninstalled.
3. Finish the installation.
4. Deploy JAMS AgentX from the JAMS Scheduler Server to the Linux Machine, or install JAMS AgentX directly onto the Linux Machine. See AgentX Documentation for detailed instructions on deploying JAMS AgentX.

NOTE: Un Checking components in the Feature Selection step will uninstall those components.

NOTE: To deploy and use JAMS AgentX, a user must be configured in JAMS with permissions to the relevant Netezza or Linux box where AgentX will be used.

Creating Netezza Jobs in JAMS
JAMS Offers NetezzaLoad, NetezzaBackup, and NetezzaSQL Execution Methods to automate Netezza.
The NetezzaLoad Execution Method allows users to automate loads to the Netezza Server.
The NetezzaBackup Execution Method allows users to automate backups of Netezza Databases.
The NetezzaSQL Execution Method allows users to run a SQL Query on a Netezza system.

Creating NetezzaLoad Jobs in JAMS
1. Create a Job in the desired folder and choose the NetezzaLoad Execution Method.
2. Select the desired folder for the NetezzaLoad Job, then ensure the Job Definitions tab is selected.
3. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
4. In the dialog, give the new Job a Name, Description (optional), and Execution Method. In this case, select NetezzaLoad to create a Netezza Load Job.
5. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
6. The Job Definition dialog will open.
   NOTE: An Execute As property should be configured with a user that has access to Netezza.
8. On the Source tab, define the available properties as follows:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Certificate File</td>
<td>Specifies the pathname of the root CA certificate file on the client system. This argument is used by Netezza clients who use peer authentication to verify the Netezza host system. The default value is NULL, which skips the peer authentication process.</td>
</tr>
<tr>
<td>Security</td>
<td>Specifies the security level that should be used for the session. The</td>
</tr>
</tbody>
</table>
Level argument has 4 available values:
0 - preferredUnsecured - This is the default value. Specify this option when you would prefer an unsecured connection, but you will accept a secured connection if the Netezza system requires one.
1 - OnlyUnsecured - Specify this option when you want an unsecured connection to the Netezza system. If the Netezza system requires a secured connection, the connection will be rejected.
2 - preferredSecured - Specify this option when you want a secured connection to the Netezza system, but you will accept an unsecured connection if the Netezza system is configured to use only unsecured connections.
3 - onlySecured - Specify this option when you want a secured connection to the Netezza system. If the Netezza system accepts only unsecured connections, or if you are attempting to connect to a Netezza system that is running a release prior to 4.5, the connection will be rejected.

Table
Specifies the table name. You can specify a fully qualified name for this value.

Port
Specifies the port to use, allowing users to override the default. Leave 0 to use the default value.

Login Timeout
Allows you to enter a different value, expressed in seconds, for the login timeout. This overrides the default value of 30 seconds.

User
Database username. [NZ_USER] by default.

Password
Database password. [NZ_PASSWORD] by default.

Database
Database name. [NZ_DATABASE] by default.

Host
Netezza hostname or IP address. [NZ_HOST] by default.

Delimiter
Specifies field-delimiter character.

Control File
Specifies the control file.

Data File
Specifies the datafile to load.

Log File
Specifies the log file name. If the file exists, this appends to it.

Bad/Rejected Rows File
Specifies the bad/rejected rows filename (overwrite if the file exists).

Output Directory
Specifies the output directory for the log and bad/rejected rows files.

Log File Size
Session variable (LOAD_LOG_MAX_FILESIZE) that specifies the size (in MB) of the log and bad/rejected rows files. The default is 2000MB (2GB).

File Buffer Size
Specifies the chunk size (MB for fileBufSize or bytes for fileBufByteSize) at which to read the data from the source file. Leave 0 to use default value.

Allow Replay
Session variables (LOAD_REPLAY_REGION and MAX_QUERY_RESTARTS) that specify the number of query restarts for load continuation if a SPU has been reset or failed over. If n is a valid non-zero number, it specifies the number of allowable query restarts. If n is 0, the system defaults to the postgres default setting.

9. Save and Close the Job definition.

Creating NetezzaBackup Jobs in JAMS

1. Select the desired folder for the Netezza Backup Job, then ensure the Job Definitions tab is selected.
2. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a **Name**, **Description** (optional), and **Execution Method**. In this case, select **NetezzaBackup** to create a Netezza Backup Job.
4. By default, the full **Job Definition** dialog will open when the Job is initially saved. Click **OK**.
5. The **Job Definition** dialog will open.
6. Define Elements, Parameters, Security, Properties, and Documentation as desired. **NOTE:** An **Execute As** property should be configured with a user that has access to Netezza.
7. On the **Source** tab, define the available properties as follows:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Database username. [NZ_USER] by default.</td>
</tr>
<tr>
<td>Password</td>
<td>Database password. [NZ_PASSWORD] by default.</td>
</tr>
<tr>
<td>Database</td>
<td>Database name. [NZ_DATABASE] by default.</td>
</tr>
<tr>
<td>Host</td>
<td>Netezza hostname or IP address. [NZ_HOST] by default.</td>
</tr>
<tr>
<td>Directory</td>
<td>Specifies a list of one or more space-separated, full pathnames of the directories where the data is to be stored. This option applies to filesystem only (not veritas or tivoli). Users can specify up to 16 directories.</td>
</tr>
<tr>
<td>Directory File</td>
<td>Specifies a file with a list of backup target directories, one per line.</td>
</tr>
<tr>
<td>Connector</td>
<td>Names the connector to which you are sending the backup, either filesystem, veritas, or tivoli.</td>
</tr>
<tr>
<td>Connector Arguments</td>
<td>Specifies a colon-separated list of passthrough arguments for the connector.</td>
</tr>
<tr>
<td>Backup Type</td>
<td>Specifies backup type.</td>
</tr>
<tr>
<td>Backup Users</td>
<td>Backs up all users and groups, and any privileges which were defined in the system database. Note that the command backs up all users and groups regardless of whether they are referenced by any permission grants. The system backs up all global adminlevel permissions (those that are not associated with particular databases). The system does not back up permissions associated with specific databases, because they are saved during regular database backup of each database.</td>
</tr>
<tr>
<td>Backup Schema Only</td>
<td>Saves only the schema of the specified database, but not the user data in tables or views. The schema includes the definitions of objects such as tables, views, synonyms, sequences, and others, as well as any access privileges defined in the database. This option is an easy way to replicate an empty database schema within a Netezza system.</td>
</tr>
<tr>
<td>Number of Streams</td>
<td>Backs up the data using the specified number of streams.</td>
</tr>
<tr>
<td>Backup Set ID</td>
<td>Specifies the backup users with to user for incremental backup, rather than the default. Note: the default backup set is the most recent backup set of the database you specify. You can override the default by using this option.</td>
</tr>
<tr>
<td>Secret Keyword</td>
<td>Specifies a string value needed to generate a 256-bit symmetric key, which is used to encrypt the host key in the backed up data.</td>
</tr>
</tbody>
</table>

8. Save and Close the Job definition.

**Creating NetezzaSQL Jobs in JAMS**

1. Create a Job in the desired folder and choose the **NetezzaSQL** Execution Method.
2. Select the desired folder for the Netezza Job, then ensure the Job Definitions tab is selected.
3. Click the **Add** button from the Control Bar to open the **Add a New JAMS Job Definition** dialog.
4. In the dialog, give the new Job a **Name**, **Description** (optional), and **Execution Method**.
In this case, select NetezzaSQL to create a Netezza SQL Job.

5. By default, the full Job Definition dialog will open when the Job is initially saved. Click **OK**.

6. The Job Definition dialog will open.

   **NOTE:** An Execute As property should be configured with a user that has access to Netezza.

8. On the Source tab, define the available properties as needed:

   **Property** | **Description**
   --- | ---
   User | Database username. [NZ_USER] by default.
   Password | Database password. [NZ_PASSWORD] by default.
   Database | Database name. [NZ_DATABASE] by default.
   Host | Netezza hostname or IP address. [NZ_HOST] by default.
   Query | Specifies SQL query to run.
   Query File | Execute queries from this file.
   Unaligned Table Output | Specifies unaligned table output mode. Normally, output is well formatted with white space added so that columns align. If you use this option, the nzsql command removes extra white space. Use this option when you want to compare results between two systems or when you want to transfer data.
   Print Rows Only | Normally the nzsql command includes column headings and a summary row for all SQL queries. Use 'Print Rows Only' option to eliminate the column headings and summary row. Use this option with the 'Unaligned Table Output' option to produce data in a transportable format.
   Expand Table Output | Users may use this option to display the query results vertically instead of in the default table/2D grid.
   HTML Table Output | Format the nzsql output to use HTML tags.
   Field Separator | Set the field separator.
   Output File | File to send the output to.

9. **Save and Close** the Job definition.
Paladin Setup and Configuration Guide

Paladin is a powerful event alerting application that integrates with your organization’s network. Use this guide to get specific details on how to download, register and configure Paladin to define a customized and effective monitoring and alerting environment for your organization.

In This Topic:
- Installing and Configuring Paladin
  - Registering Paladin
  - Completing the Paladin Profile
  - Downloading Certifications
- Paladin Navigation
  - Viewing and Editing Objects
- Security
- Managing Profiles and Inviting Users
- Monitors
- Groups
- Response Teams
- Resolving Events
- Setting up the Paladin Alert in JAMS

Installing and Configuring Paladin

To get Paladin up and running, users at minimum need to install Paladin, Authenticate and Complete their User definition, and Download and run the Paladin Certificate Installer on the JAMS Server.

Installing JAMS and Paladin

To get started, you will need to download and install JAMS. After confirming the Paladin license purchase with your account representative, download the production ready JAMS installer from the support site.

Next, Install JAMS and make sure that the JAMS Integration Pack for Paladin Event Handling has been installed. If not, re-run the installer and enable the checkbox for Paladin on the Integration Packs step. Note that unchecking any previously installed components will uninstall those items. For a detailed information on how to install JAMS, please refer to the JAMS Installation Guide.
Registering Paladin

As a part of the purchase, your company profile is created by the account representative, triggering an email invitation to register Paladin.

1. Open the Email Invitation and click the Paladin link.

   ![Email Invitation]
   
   Hello,
   You've received an invite to Paladin. Try it out by following this link:
   https://paladinweb.mrvsap.com/invitation/accept/11220dad-9225-49bf-a27d-19d1b668b6

2. A web page will open. Click **Select Authorization** to open the Auth0 popup and begin the authorization process for Paladin.
3. In the Auth0 popup, click the **Sign Up** tab, then select from one of the authentication methods. Users have the option of social network credentials, or using a standard Email & Password with the Sign Up button. 
In this example, Microsoft is used. Clicking the Microsoft button immediately takes the new user to the Microsoft Login page. 
**NOTE:** If the Email/Password option is used, a confirmation email will be sent to the listed email address. Users must follow the instructions in that email to verify their account.

4. Enter the **credentials** for the given authentication method, and resolve any additional dialog boxes that may appear.
5. With the verification process complete, users will be taken to the Paladin site, https://paladinweb.mvpsi.com/

6. On any first login, users should immediately complete their Paladin profile before navigating through or editing Paladin.

**Completing the Paladin Profile**

The first action any user should take in Paladin is to complete their Paladin Profile. Failure to complete the Paladin Profile may result in errors.

1. From the Paladin Site, click the **Vertical Ellipses** in the upper right-hand corner. A drop-down will appear.
2. Select the **Edit Profile** option from the drop-down list.
3. A Profile Panel will appear. Click the Pencil icon to begin editing.

4. Set Profile options as desired. At minimum, Name, E-Mail, and Time Zone are required for initial Paladin Profile configuration. See the Managing Profiles and Inviting Users section for more information on Profile Properties.

5. With options set as desired, use the Save button to commit the changes.
Installing the Paladin Certificate

Paladin must be able to send and receive messages on port 443 to function properly. Ensure port 443 is open, then follow the instructions below to download and install the Paladin certificate.

1. Navigate to the Paladin Home page, then click on **Select a Certificate** in the **Quick Actions** section.
2. Select the certificate from the pop-up list, then use the **Download** button to download the Certificate Installer.
3. Open the **PaladinApiCertificateInstaller** from the download location.  
   **NOTE:** Users MUST run the installer on the JAMS Server.

4. Read and accept the License Agreement, then click **Install**.  
The Installer will add the Paladin certificate to the Local Computer's Personal Certificates.

5. Once the installer has completed, click **Finish**.
6. Open the JAMS Client as an administrator on the JAMS Server.  
   **NOTE:** If the JAMS Client was open when the certificate was installed, restart the JAMS Client.

7. Open the Configuration shortcut, then select the Integrations tab to view the Paladin Integration Properties.

8. Set the Certificate property to the Friendly Name of the Paladin Certificate, "ApiPFXCertificate_wixCert_0". Users that attempt to rename the Paladin Certificate must use their custom Friendly Name as the Certificate value.

9. **Save** the Certificate value change. **Do Not edit any other Paladin Properties.**

10. JAMS is now ready to send Paladin alerts through the use of **Paladin Event Elements.**

**Paladin Navigation**

The Navigation Panel, located on the left side of the Paladin UI, contains a set of shortcuts for each Paladin
module. Module shortcuts are displayed based on the Role assigned to the logged in user.

- **Home** is the splash screen for Paladin, and includes preconfigured items:
  - *Quick Actions* - If users have proper permissions, they can invite new users directly from the Quick Actions Bar with the Invite User button.
  - *Event Summary* - A list of currently active events. Users with proper permissions may resolve events directly from the Event Summary section.
  - *Recent Events* - A graphical display of events that recently occurred, sorted by day and by week.
  - *Current Events* - A graphical display of active events in the environment.

- **Events** is a list of any events in the environment. It can be filtered to show All, Active, or Resolved Events. Users with sufficient permissions can resolve Active Events from here.

- **Monitors** contains a list of preconfigured monitors, sorted by group. Individual Monitors can be edited from this shortcut.

- **Configuration** is only available for Administrators. It contains sections to view, add, or edit company specific configuration, including:
  - *People* - People are the individuals in a company. The People shortcut allows Admins to view, add, or edit any people within their company.
Groups - Groups are used to logically organize other objects, including People. The Groups shortcut allows Admins to view, add, or edit any Groups within their company.

Response Teams - Response teams are logical assortments of People or Groups used to determine who will be notified when an event occurs. The Response Teams shortcut allows Admins to view, add, or edit any Response Teams within their company.

Status Manager - View and update the status of Response Team members to determine availability to receive notifications.
Viewing and Editing Objects

General Viewing and Editing

Most objects in Paladin are edited the same way. The instructions below outline general editing practices for all objects in Paladin.

Note that a user’s ability to edit a given object depends on their role.

1. Navigate to the module containing the data that will be viewed or edited.
2. The Module will contain a list of items, with high-level information about each item. If the user has the required permissions, there will also be a button to Add a new item.
3. Click on a specific item in the list to view the details of that item. If the user has the required permissions, there will also be an Edit button (pencil icon).
4. When editing, an Undo button (curved arrow) and Save button (floppy disk) will appear. The Save button will preserve any changes made so far. The Undo button will revert unsaved changes.
5. When satisfied with the changes, click Save, then close the item.

Security

Security in Paladin is comprised of Roles and Permissions. Role options currently include User and Admin. An overview of Role Permissions is available in the table below.

<table>
<thead>
<tr>
<th>Permission</th>
<th>User</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive Notifications</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Resolve Ongoing Events</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Read and access all of the company’s settings and objects</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Update their own user profile</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Edit all the company’s settings and objects</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Create new objects</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Invite new users to the company</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Managing Profiles and Inviting Users

In Paladin, Users have a set of properties that should be defined to properly configure them.

- **First and Last Name** are the first and last name defined for the given user. A maximum of 60 characters may be used.
- **Primary E-mail** is the defined user’s primary contact email.
- **Primary Time Zone** defines the local time zone for the given user.
- **Quite Time Begin** and **Quiet Time End** are used to create a window during any given day where Paladin alerts should not be sent to the user.
- **Off Duty / On Duty Toggle** is used to define whether the given user is considered “on Duty” and available to receive Paladin notifications.

*NOTE:* Users must be On Duty to receive Paladin Alerts.

- **Communication Preferences** are the defined Email Address, Phone Number, and Cell Phone Number associated with a given user.
  - **Email Address** is defined as a valid email address. The email type must also be defined as either Text or HTML. Users may also define a Severity associated with the Email Address.
  - Phone Number is the defined landline or office phone of the given user and entered using the E.164 format as shown below. Users may also define a Severity associated with the telephone number.
  - Cell Phone Number is the defined Cell Phone associated with the given user. The defined Cell Phone Number should be able to receive text messages. User may set an associated Severity Level with the Cell Phone number. The Cell Phone number must use the E.164 format as shown below.
  - **Formatting Phone and Cell Phone Numbers** must use E.164 Formatting.
    - E.164 Formatting utilizes the international telephone and cell phone numbering plan to ensure each device has a unique number.
    - There is a maximum of fifteen (15) characters.
    - A formatting example can be seen below:

    ![Communication Preferences](image)

- **Response Teams** are the response teams that the defined user should belong to. Users may belong to any number of Response Teams. To define a Response Team, use the Response Team Search function and select the desired Response Team from the drop-down results list.
- **Groups** are logical collections of users. A user may belong to any number of Groups. To add a Group,
Inviting a user to a Company

1. While Logged in as an Administrator, Navigate to the Home Module.
2. Click the Invite User button in the Quick Actions Bar. A panel will appear.
3. Enter the Email Address of the user to invite.
4. Select the Role that will be assigned to the new user - User or Admin.
5. Click the Send Invitation button.
6. When the invitation has been successfully sent, a confirmation panel will appear.
7. The newly invited user will need to complete the authentication and profile configuration steps seen in the Registering Paladin and Configuring the Paladin Profile instructions.

Monitors

Monitor Properties

Monitors in Paladin have a standard set of defined Properties:

- **Monitor Group** defines which group the new monitor belongs to. In nearly all environments, monitors will reside in the JAMS Monitor Group.
- **Monitor Type** defines the type of monitor that is being created. For nearly all environments, all monitors will...
be the JAMS Monitor Type.

**Monitor Name** is the friendly name used to reference the monitor.

**Description** is the long-text reference of the monitor.

**Default Severity** for new events is the severity that will be assigned to events coming into the monitor when no severity was set on the Paladin Element. Severity levels range from 1 to 10.

**Event Escalation** is a slider that defines how often Event severity should be raised for the given monitor.

**Primary Response Team** is the response team that should be responsible for responding to events on the given monitor.

### Creating Monitors

1. Navigate to the Monitors section, then use the **New Monitor** button to open the create panel.

2. On the Create panel, enter the properties as necessary, using the **JAMS Monitor Group** and **JAMS Monitor Type**.

3. When the properties have been set as desired, **Save and Close** the create panel.
**Groups**

Groups are collections of objects and may include any number of defined People, Groups, or Response Teams.

**Group Properties**

- **Group Name** is the friendly name used to reference the group.
- **Description** is the long-text reference of the group.

**Creating Groups**

1. Navigate to the Groups section, then use the New Group button to open the create panel.
2. On the Create panel, enter the properties and add People, Groups, or Response Teams as necessary.

3. When the properties have been set as desired, save and close the create panel.
Response Teams

Response Team Properties

Response Team Name is the friendly name used to reference the Response Team.

Description is the long-text reference of the monitor.

Ignore on Duty? is a toggle used to determine whether the Response Team should ignore the “On Duty” status set at the user level.

Ignore Quiet Time? is a toggle used to determine whether the Response Team should ignore the “Quiet Time” set at the user level.

Escalation Response Team is a secondary response team that will be notified if the given response team does not resolve an event.

Notification Interval defines how often notifications should be repeated.

Notification Repeat Count defines how many times notifications should repeat on the defined Notification Interval.

Minimum Notification Severity is the minimum Event Severity that will trigger notifications to the response team.

Creating Response Teams

1. Navigate to the Response Teams section, then use the New Response Team button to open the create panel.
2. On the Create panel, enter the properties as necessary.

3. When the properties have been set as desired, save and close the create panel.
Resolving Events

Paladin’s new UI allows users to resolve events directly from the Paladin Web App. To Resolve Events:

From the Home Module

1. Navigate to the Event Summary in the Home Module.
2. Find the event in the Event Summary list and click the Resolve button.
3. The Event will be resolved with the default messages of “Manually resolved by <<user>>.” The current date and time are recorded on the resolution.

From the Events Module

1. Navigate to the Events Module.
2. Find the event that should be resolved and click it. A panel will appear.
3. Toggle the message to either default or custom as desired.
4. Click the Resolve button.
5. If the event was resolved with the custom message option set, a panel will appear for entering the custom message.
Setting up the Paladin Alert in JAMS

Users must set up the Paladin Alerts in JAMS. To set up these Alerts, a Paladin Event Element is added to a Job or Folder.

Configuring a Paladin Alert

1. Open the Properties dialog for the Job or Folder where the Paladin alert will be added.
2. Click on the Elements tab, the use the Add... button to open the Add Element dialog.
3. Select Paladin Event from the EventHandler section, then click Next >.
4. Set the Element Properties as desired, then click Finish.
5. Save & Close the Job or Folder.

Paladin Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor ID</td>
<td>The Monitor(s) that should receive the event. Use a comma separated list to define multiple Monitors.</td>
</tr>
<tr>
<td>Severity</td>
<td>The defined severity of this event. If left as 0, the severity will be set to the Monitor's default severity.</td>
</tr>
<tr>
<td>Summary</td>
<td>A summary of the event.</td>
</tr>
<tr>
<td>Detail</td>
<td>A detailed description of the event.</td>
</tr>
<tr>
<td>Ask for a response</td>
<td>Should JAMS ask for a response from Paladin and take action based on the response?</td>
</tr>
<tr>
<td>Upload the Log file</td>
<td>Should the Job's log file be uploaded to Paladin?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Events</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Success</td>
<td>Defines whether the event should be triggered when the entry completes with a successful completion severity.</td>
</tr>
<tr>
<td>Entry Informational</td>
<td>Defines whether the event should be triggered when the entry completes with an informational completion severity.</td>
</tr>
<tr>
<td>Entry Warning</td>
<td>Defines whether the event should be triggered when the entry completes with a warning completion severity.</td>
</tr>
<tr>
<td>Entry Error</td>
<td>Defines whether the event should be triggered when the entry completes with an error completion severity.</td>
</tr>
<tr>
<td>Entry Fatal</td>
<td>Defines whether the event should be triggered when the entry completes with a fatal completion severity.</td>
</tr>
<tr>
<td>Entry Started</td>
<td>Defines whether the event should be triggered when the Job has started executing.</td>
</tr>
<tr>
<td>Normal Event</td>
<td>Defines whether the event should be triggered when a normal event occurs.</td>
</tr>
<tr>
<td>Low Event</td>
<td>Defines whether the event should be triggered when a low event occurs.</td>
</tr>
<tr>
<td>Moderate Event</td>
<td>Defines whether the event should be triggered when a moderate event occurs.</td>
</tr>
<tr>
<td>Event Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>High Event</td>
<td>Defines whether the event should be triggered when a high event occurs.</td>
</tr>
<tr>
<td>Urgent Event</td>
<td>Defines whether the event should be triggered when an urgent event occurs.</td>
</tr>
<tr>
<td>Critical Event</td>
<td>Defines whether the event should be triggered when a critical event occurs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>A toggle used to enable or disable the Event Handler.</td>
</tr>
</tbody>
</table>
PeopleSoft Integration Pack

The JAMS Integration Pack for PeopleSoft allows users to initiate PeopleSoft processes and take advantage of PeopleSoft’s monitoring and report distribution capabilities using the sophisticated and reliable JAMS scheduling software.

The JAMS Integration Pack acts as an intermediary between JAMS and PeopleSoft. When the adapter runs it initiates communication with the PeopleSoft Process Scheduler and instructs PeopleSoft to execute the PeopleSoft jobs as configured in JAMS. The process can then be monitored, canceled and restarted in either PeopleSoft or JAMS.

Supported Process Types

The JAMS Integration Pack for PeopleSoft can initiate the following process types delivered by PeopleSoft:

- Application Engine
- COBOL SQL
- CRW Online
- Crystal
- Crystal Check
- nVision
- nVision-Report
- nVision-ReportBook
- PS Job
- SQR Process
- SQR Report
- XML Publisher

In addition, the JAMS Integration Pack for PeopleSoft can initiate custom process types that you have created. For best results, these custom processes must be “API Aware”.

For more information, please refer to the PeopleSoft Process Scheduler PeopleBook under "Using Process Request APIs".

Supported PeopleTools Versions

The JAMS Integration Pack for PeopleSoft works with PeopleTools versions 8.48 or newer.

To request certification for additional PeopleTools releases, please contact your JAMS customer service representative.

Supported Operating Systems

The JAMS Integration Pack for PeopleSoft works with any Unix/Linux or Windows operating system that is supported by Oracle.

Note: the Windows operating system must run Microsoft PowerShell.

JAMS Integration Overview and Process Flow

The JAMS Integration Pack for PeopleSoft consists of two parts:

1. The JAMS components that capture information about any PeopleSoft processes that you want to schedule and . . .
2. An Application Engine process called MVP_SCHDPRECS that schedules the jobs through the PeopleSoft Process Scheduler API, monitors and cancels the job when necessary.

The following diagram illustrates the detailed process flow of the JAMS Integration for PeopleSoft.
Installing the JAMS Integration Pack for PeopleSoft

The JAMS Integration for PeopleSoft contains an Application Engine program and supporting app designer objects to schedule the jobs. These objects are included in a project that must be installed and operating in the PeopleSoft environment.

1. Launch the Application Designer and connect to your PeopleSoft instance in 2-tier Mode.
2. Navigate to Tools > Copy Project > From File.
3. From your JAMS Scheduler machine, copy the `MVP_JAMS_ADDON_FOR_PS.ini` and `MVP_JAMS_ADDON_FOR_PS.xml` files from the `Program Files\MVPSI\JAMS\Scheduler` directory. Paste them into a folder named, `MVP_JAMS_ADDON_FOR_PS` that is manually created on the PeopleSoft server.

   Browse to the location of that newly created folder. The Project name “MVP_JAMS_ADDON_FOR_PS” appears in the lower box. Click the `Select` button.

4. The Copy from File dialog box appears. Select the `Copy` button.
5. Set the default tablespace.

The JAMS Integration for PeopleSoft requires one table to maintain its scheduling process. Your DBA will likely want to specify which tablespace to utilize, depending on your organization's database platform.

Set the tablespace using the steps below:

- Expand the “Records” node in the project tree.
- Double-click **MVP_SCHD_LOG** to open the record definition.
- Navigate to Tools > Data Administration > Set Tablespace. The Change Space window appears.
- Select the appropriate Platform and Tablespace Name depending on your organization’s database platform and click **OK**.
- Save the record.
6. Build the Project to create the `PS_MVP_SCHD_RCNTL` table:
   - Navigate to `Build > Project`.
   - Select the `Create Tables` and `Create Indexes` checkboxes.
   - Choose the `Execute SQL Now` button.
   - Select the `Build` button
   - To complete the installation process, review the log file for errors.

### Configuring the Execution Method

Before JAMS can execute a Job, additional information about your PeopleSoft environment will need to be provided. To get started, follow the steps below.

1. Open the JAMS Client as an administrator.
2. Select the `Execution Methods` shortcut from the Menu.
3. Locate the listed PeopleSoft Job Execution Methods.
   - If your organization’s PeopleSoft Process Scheduler runs on Unix or Linux, open the `PeopleSoftJobSsh`.
   - If your PeopleSoft Process Scheduler runs on Windows, open `PeopleSoftJobWindows`.
4. When the definition dialog opens, navigate to the `Parameters` tab.
5. Locate the parameter `PS_ORACLE_HOME`. Open the parameter definition dialog by double clicking the parameter or highlight it and choose the `Properties` button.
6. Enter the correct information in the Default Value field.
8. Click the right arrow on the window header to move to the next parameter. Define parameters as necessary to connect to the PeopleSoft environment.
9. Save and Close the Parameters dialog
10. Save and Close the Execution Method dialog.

**Note:** All but the "Additional_Param_#" parameters must be reviewed. Below is a reference table that describes each parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBInstance</td>
<td>Database Instance/Name</td>
<td>EP91DMO</td>
</tr>
<tr>
<td>DBType</td>
<td>Database Type</td>
<td>ORACLEMSSQLDB2SYBASE</td>
</tr>
<tr>
<td>PS_ORACLE_HOME</td>
<td>Oracle Home</td>
<td>/u01/app/oracle/product/11.1.0/db_1</td>
</tr>
<tr>
<td>PS_ORACLE_SID</td>
<td>Oracle Database Name</td>
<td>EP91DMO</td>
</tr>
<tr>
<td>PS_TUXDIR</td>
<td>Tuxedo Directory</td>
<td>/usr/local/psoft/Oracle/Middleware/tuxedo10gR3</td>
</tr>
<tr>
<td>PS_HOME</td>
<td>PeopleSoft Home. You can use $ORACLE_SID as a variable.</td>
<td>/usr/local/psoft/ORACLE_SID</td>
</tr>
<tr>
<td>PS_CFG_HOME</td>
<td>Configuration Home</td>
<td>/usr/local/psoft/cfg850</td>
</tr>
<tr>
<td>PS_SQRDIR</td>
<td>SQR Directory. You can use $ORACLE_SID as a variable.</td>
<td>/usr/local/psoft/ORACLE_SID/bin/sqr/ORABIN</td>
</tr>
<tr>
<td>PS_SERVER_CFG</td>
<td>Server Configuration</td>
<td>$PS_CFG_HOME/appserv/prcs/ORACLE_SID/psprcs.cfg</td>
</tr>
</tbody>
</table>
Setting Up a JAMS Job for PeopleSoft

Before setting up any Job in JAMS, the user running the process should log into PeopleSoft, navigate to the run control page and enter the information required on the PeopleSoft page.

1. Select the desired folder for the PeopleSoft Job, then ensure the Job Definitions tab is selected.
2. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a Name, Description (optional), and Execution Method. 
   NOTE: Select PeopleSoftJobWindows or PeopleSoftJobSsh to create a PeopleSoft Job. If the organization's PeopleSoft Process Scheduler runs on UNIX or Linux, choose the PeopleSoftJobSsh Execution Method. If the Scheduler runs on Windows, select the PeopleSoftJobWindows Execution Method.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
5. The Job Definition dialog will open.
   NOTE: An Execute As property should be configured with a user that has access to the PeopleSoft Server.
7. On the Source tab, populate the Run Control ID with the one that is already set up in PeopleSoft.
8. Enter the name of the PeopleSoft User who owns the Run Control.
9. Choose the Process Type from the dropdown and enter the Process Name.
10. Select the Output Type and Output Destination Format using the individual dropdowns. The Output Destination is required only if the selected Output Type is File.

Working With Crystal Report Jobs

Crystal Reports jobs in PeopleSoft require that Query Bind Variables be passed to the job on the command line. If scheduling a Crystal Report, use the following steps:

1. If the PeopleSoft process requires additional parameters, you must add these using the Parameters tab. Most Crystal Reports in PeopleSoft require additional parameters. Use the parameter name Additional_Parm_x for the name. For parameters 1 through 9, there is no leading zero. See the examples below to identify which parameters to use.
2. Running the Crystal Report inside of PeopleSoft is a great way to identify which parameters need to be passed. Once it has been run, open the process in the PeopleSoft Process Monitor (PeopleTools > Process Scheduler > Process Monitor).
3. Locate the process (Crystal in most cases) and select the Details link.
4. Click the Parameters link.
5. Locate the **Command Line** and review all data, especially at the end of the command line. Most of the time the required parameters come after the “ORIENTL” parameter. In the following example, the five required parameters are: "US001" "2007-11-07" "2011-11-07" "VP1" "posted_voucher_listing". When adding these parameters into JAMS make sure to exclude the double quotes.

6. In the following example, the Crystal APY1020 requires additional parameters.
7. Here's an example of Parameter 1. Notice the Default Value is US001 without the double quotes.

8. Create the remaining parameters (2007-11-07 "2011-11-07" "VP1" "posted_voucher_listing").
9. Save and Close the Parameters dialog.
10. **Save and Close** the Job Definition.

11. When the JAMS Job is submitted, the five parameters will appear (as long as the parameters are not hidden).

    **Note:** The additional parameters do not necessarily have to be populated when creating the JAMS Job, but the parameter name itself must be set up at that time.
Running SAS Jobs with JAMS

**Windows Command Procedure Method**

JAMS can execute SAS programs using the Windows Command Procedure Execution Method.

Below is an example of an SAS job-source:

```
c:\YourSASPath\sas.exe
  -autoexec d:\sas\autoexec.sas
  -config d:\sas\config.sas
  -nologo
  -log d:\work\logs\pull_sample200102131244.log
  -sysin d:\YourDir\YourProgram.sas
```

**SAS Execution Method**

As an alternative, you can create a SAS execution method from within JAMS. This allows the source tab of a Job definition to contain the contents of your SAS program.

1. Select the Execution Methods shortcut from the Menu.
2. Click the Add button from the Control Bar to open the Add an Execution Method dialog.
3. In the Add an Execution Method dialog, give the Execution Method a Name, Description (optional), and Base Method.
   **NOTE:** Use the Command Base Method for SAS.
4. Click OK. By default, JAMS will open the full Execution Method Definition dialog when the Execution Method is initially saved.
5. Edit the Parameters, Template, Properties, and Job Properties as necessary.

**JAMS VBScript Execution Method**

SAS Projects (EGP files) can be saved as Visual Basic scripts from the SAS Enterprise Guide and executed using the JAMS VBScript Execution Method.
Using SAP Execution Methods to Run JAMS Jobs

JAMS offers two Execution Methods that support job execution processes using SAP outside of the SAP Business Objects Data Services Integration Pack.

- **SAPJobV2** acts as a replacement for the current **SAPJob** and **SAPStandardJob** Execution Methods.
- **SAPProcessChain** provides process chain execution handling in JAMS

To use the SAP Process Chain and SAP Job Execution Methods, users must have the following:

- SAP ERP Central Component (ECC) 6.0, with SAP Enhancement Package 7 (EHP) or newer
- SAP NetWeaver 7.4 or newer
- SAP Environment must be a Unicode system

The two screenshots below highlights these new Execution Methods along with their respective Parameters.
### JAMS 7.X Installation Guide

#### Method Name: SAPJobV2

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Prompt</th>
<th>Default Value</th>
<th>Help Text</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSSAPServer</td>
<td>Text</td>
<td></td>
<td>10.127.128.117</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAMSSAPClient</td>
<td>Text</td>
<td></td>
<td>001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Method Name: SAPProcessChain

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Prompt</th>
<th>Default Value</th>
<th>Help Text</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSSAPServer</td>
<td>Text</td>
<td></td>
<td>10.127.128.117</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAMSSAPClient</td>
<td>Text</td>
<td></td>
<td>001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAMISTraceLevel</td>
<td>Text</td>
<td>Select the JAMSTrace Level:</td>
<td>Off</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NOTE: To define a system number on your execution method, set the JAMSSAPServer parameter to "Servername:SystemNumber" for the given execution method. E.g. if the server was "10.127.128.117" and the System Number was "23", set the JAMSSAPServer parameter to be "10.127.128.117:23".

NOTE: SAP system numbers are used to define the listener port. The port is defined as "33XX", where "XX" is the system number. If the system number were "23", the SAP port would be "3323".

NOTE: To define language on an SAPJobV2 Job, create a parameter named JAMSSAPLanguage on the Job and set the parameter value to the language code. To define language on an SAPProcessChain Job, use the Language field in the Job Source.

Parallel Execution for JAMS and SAP Jobs

Once JAMS submits the Job, SAP then creates and executes a parallel job as shown in the following two screenshots.
Job Execution Logs

Once a Job executes, logs can be viewed from either the JAMS or SAP environments.

Authorized users may, in certain cases, cancel a currently executing Job. In SAP, such a job would go into a cancelled state. Logs can then be checked in both environments.

Running a JAMS Job using the SAPProcessChain Execution Method

NOTE: Users may set the SAP System Language on SAPProcessChain Jobs by using the Language field in the Job Source.
When the user creates a Job using the SAPProcessChain execution method it is then submitted and executed normally in JAMS. The Job’s output includes a process chain number (log ID). The JAMS log file then displays the output of the process chain.

Using the SAP.Net Connector with the JAMS Adaptor

These permissions describe the necessary roles and permissions in SAP to allow users to run a Job using the new JAMS adapter.

SAP_BC_BATCH_ADMIN -- This role contains all authorizations for background processing administration, including the creation of background jobs and general administrations functions (SMxx transaction codes, in particular SM36, SM37, SM50, and SM51).

SAP_BC_ENDUSER -- This role contains non-critical basis authorizations for all users, including job creation and job release.

S_BTCH_ADM -- this role allows user the ability to perform any operation on any job, including:
- Selecting jobs from all clients (from the Job Overview, Transaction SM37)
- Deleting any job
- Releasing jobs to start
- Changing jobs, including copying, canceling, checking, repeating, and capturing and debugging
- Displaying jobs, job steps, and job logs
- Triggering events manually (transaction SM64)
- Editing system events
- Working with raised events in the event history (transaction SM62)
- Using restricted job classes A or B
- Scheduling an external program in a job

Five User Type are available in SAP. And as per SAP recommendations, the User should be “System”.

<table>
<thead>
<tr>
<th>Alias</th>
<th>User Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Dialog</td>
</tr>
<tr>
<td>Security Policy</td>
<td>Dialog</td>
<td>System</td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td>Communications Data</td>
</tr>
<tr>
<td>Password Status</td>
<td>Reference (Logon not possible)</td>
<td>Service</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialog</td>
<td>Individual, interactive system access.</td>
</tr>
<tr>
<td>System</td>
<td>Background processing and communication within a system (such as RFC users for ALE, Workflow, TMS, and CUA).</td>
</tr>
<tr>
<td>Communication</td>
<td>Dialog-free communication for external RFC calls.</td>
</tr>
<tr>
<td>Service</td>
<td>Dialog user available to a larger, anonymous group of users.</td>
</tr>
<tr>
<td>Reference</td>
<td>General, non-person related users that allows the assignment of additional identical authorizations, such as for Internet users created with transaction SU01. No logon is possible.</td>
</tr>
</tbody>
</table>

Overview of the properties of users with different user types.

<table>
<thead>
<tr>
<th>Property \ User Type</th>
<th>Dialog</th>
<th>Communication</th>
<th>System</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialog Logon (SAP GUI)</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>Multiple Logons</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RFC Logon</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Property \ User Type</td>
<td>Dialog</td>
<td>Communication</td>
<td>System</td>
<td>Service</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Background Job Execution</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Password Change</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Logon Ticket can be Generated</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

"X" represents Yes
Run SAP Business Objects Data Services in JAMS

The SAPDataService execution method allows users to automate any job that already exists in an SAP Business Objects Server environment.

In This Topic:

- Installing the SAP Business Objects Data Services Integration Pack
- Configuring the SAPDataServices Execution Method
- Creating SAP Business Objects Data Services Jobs in JAMS

Installing the SAP Data Services Integration Pack

1. Run the JAMS Installer
2. On the Feature Selection step, ensure the JAMS Integration Pack for SAP Data Services component is checked
3. Complete the installation

**NOTE**: Un checking components will uninstall those components.

Configuring the SAP Data Service Execution Method

1. Run the JAMS Client as an administrator.
2. Select the Execution Methods shortcut from the JAMS Menu.
3. Double-click the SAPDataService execution method to open the Execution Method Definition dialog.
4. Select the Parameters tab, then double click on any parameter to open the Parameter Definition dialog.
5. Set the SAPAuthentication and SAPEndpoint values to reflect the given environment.
6. Save and Close the Parameters dialog.
7. Save and Close the Execution Method dialog.

**NOTE:** The user running SAP Data Service Jobs must have full access to the SAP BODS server.

### Creating an SAP Data Services Job in JAMS

1. Select the desired folder for the SAP BODS Job, then ensure the Job Definitions tab is selected.
2. Click the **Add** button from the Control Bar to open the *Add a New JAMS Job Definition* dialog.
3. In the dialog, give the new Job a **Name**, **Description** (optional), and **Execution Method**.
   In this case, select **SAPDataService** to create an SAP Business Objects Data Services Job.
4. By default, the full **Job Definition** dialog will open when the Job is initially saved. Click **OK**.
5. The **Job Definition** dialog will open.
   **NOTE:** An **Execute As** property should be configured with a user that has access to the SAP BODS Server.
7. Open the **Source** tab to set the SAP BODS Job options.
8. Use the drop down list to select a CMS **Authentication** method for your environment.
9. Use the **Repository** drop-down list to select the relevant directory where the SAP Job is stored.
10. Use the **Job Name** drop-down list to select the desired SAP Job from the specified **Repository**.
11. When the relevant SAP Job has been configured in the new JAMS Job, **Save and Close** the Job definition.
Symitar in JAMS

In This Topic:
- Installing the Symitar Integration Pack
- Creating a Symitar Job
- Defining a Symitar Job using the Workflow Interface
- Symitar Activities

Installing the Symitar Integration Pack
If the JAMS Integration Pack for Symitar is not installed, run the JAMS Installer and check the "JAMS Integration Pack for Symitar" component in the Feature Selection step.

NOTE: Un Checking other components will uninstall those components.

NOTE: In some installations, the host.config file must be configured in order to allow a machine to access the Symitar server.

Creating a Symitar Job
1. Create a Job in the Desired folder and choose the Workflow Execution Method
2. Select the desired folder for the Symitar Job, then ensure the Job Definitions tab is selected.
3. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
4. In the dialog, give the new Job a Name, Description (optional), and Execution Method. In this case, select Workflow to create the Symitar Job.
5. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
6. The Job Definition dialog will open.
   NOTE: An Execute As property should be configured with a user that has access to the Symitar Server.
8. Select the Source tab, and define the Job using the Workflow Interface.
   (Keep reading this document for more information on the Workflow Interface and Symitar Activities)
9. Save and Close the Job Definition dialog.

Defining a Symitar Job using the Workflow Interface

Workflow Interface
The source for a Symitar Job displays the Workflow Interface. This interface is comprised of three panels: the Toolbox panel, Activity panel, and Properties panel. Symitar Activities can be found in the Toolbox panel, under the "SYM" group. Activities will be used to define the Symitar Job.
Toolbox Panel
The Toolbox Panel contains dozens of grouped Activities that can be used in the workflow. These activities are the building blocks of a Workflow Job. Activities may be used exclusively, or combined with other activities to create a workflow.

Activity Panel
The Activity Panel is the Designer section of the workflow. Drag activities from the toolbox to this area to add them to the workflow, or move those activities around in the workflow window. Right-click on items in this panel to get additional options.

Properties Panel
Shows the available properties of any selected element or activity. The properties of each activity can be quickly set from this panel.

Building Symitar Workflows
A single Symitar Activity can be utilized by dragging one of the available activities from the Toolbox Panel into the Activity Panel and setting the properties for that Activity.

To utilize multiple activities in a workflow, JAMS requires a Parent Activity to contain other activities. Note that JAMS will automatically add “Sequences” to the workflow if another parent activity is not set. Sequences may be nested inside of other Sequences or Parent Activities in a workflow.

Automatically Add Sequences:
1. With at least one activity in the Activity panel, drag another activity from the toolbox panel to the Activity panel.
2. Hover the new activity over any existing activity. Grey bars will appear to show possible placement locations.
3. Release the activity in the desired location. A sequence will be added if necessary. Note that moving existing activities will also cause the placement location bars to appear, and will also result in the creation Sequences when necessary.

Passing Parameters to a Symitar Job
Typically, executing a Symitar job involves responding to a series of prompts; for example, what is the job name?. . . the queue?, etc.
A Symitar job may include other prompts that are unique to the job. When starting a Symitar job, if JAMS encounters an unexpected prompt, it looks for a workflow variable/JAMS parameter named "SymParam1". If this parameter is found, JAMS inserts its value as the response to the prompt. However, if this variable/parameter is not found, JAMS uses the default value defined in the Symitar job.

**Note:** When looking for a "SymParam", JAMS searches first for a Workflow variable with that name. If that variable doesn’t exist, it then looks for a JAMS Parameter with that name. Each time JAMS looks for a parameter it increments the number, so the second parameter would be “SymParam2”, etc.

## Symitar Activities

A small overview of properties and function of each Symitar Activity are listed here. When defining Symitar Activities, it is important to keep in mind:

- Symitar pathways will be case sensitive in UNIX or LINUX environments
- DateTime values are formatted as YYYYMMDD

### DataFiletoPC

Downloads a DataFile from the FTP Server to the PC. DataFile is downloaded from the SYMnnn/DATAFILES directory, where nnn is a three-digit SYM Number (e.g., ’007’).

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>The local directory is the download location for the SYM DataFile.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Use to modify the property name as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>The AIX password. Use when overriding the value in the SymConfig Activity.</td>
</tr>
<tr>
<td></td>
<td>As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (4) enter the password for this account, and (5) then use that JAMS User Name in the JAMSUserName workflow property.</td>
</tr>
<tr>
<td>ServerFileName</td>
<td>The name of the DataFile on the SYM Server in the '/SYMnnn/DATAFILES' directory.</td>
</tr>
<tr>
<td>Servername</td>
<td>The name of the AIX server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>The Sym Number. Use when overriding the value in the SymConfig activity. This parameter defaults to ’000’.</td>
</tr>
<tr>
<td>UserName</td>
<td>The AIX UserName. Use when overriding the value in the SymConfig activity.</td>
</tr>
</tbody>
</table>

### DeleteLetterFile

Deletes a LetterFile from the ’/SYMnnn/LETTERSCPECS’ directory

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Use to modify the property name as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>The AIX Password (Use when overriding the value in the SymConfig Activity).</td>
</tr>
<tr>
<td></td>
<td>As an alternative, if you wish to hide the password from viewing, create a User in JAMS, set the &quot;Logon As&quot; to the AIX Username, click the password button and enter the password for this account, and then use that JAMS User Name in the JAMSUserName workflow property.</td>
</tr>
</tbody>
</table>
| ServerFileName| The name of the LetterFile to delete from the ’/SYMnnn/LETTERSCPECS’ directory.
Directory on the SYM Server.

**ServerName**
The name of the AIX Server. Use when overriding the value set in the SymConfig activity.

**SymNumber**
The Sym Number. Use when overriding the value in the SymConfig Activity. The parameter defaults to '000'.

**UserName**
The AIX UserName. Use when overriding the value in the SymConfig activity.

---

**EmailReports**
Send a report generated from ReportFileToPC as an email attachment. The report must be assigned to a Workflow Variable of type SymReport.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCCAddress</td>
<td>The address of user(s) to receive blind carbon copies of the email.</td>
</tr>
<tr>
<td>CCAddress</td>
<td>The address of user(s) to receive carbon copies of the email.</td>
</tr>
<tr>
<td>FromAddress</td>
<td>The address to be used to send the email.</td>
</tr>
<tr>
<td>Reports</td>
<td>The report(s) to be sent as attachments to the email.</td>
</tr>
<tr>
<td>Subject</td>
<td>The subject of the email to be sent.</td>
</tr>
<tr>
<td>ToAddress</td>
<td>The address of the user(s) to receive the email.</td>
</tr>
<tr>
<td>BodyMessage</td>
<td>The body of the message to be sent.</td>
</tr>
<tr>
<td>SMTPServer</td>
<td>The SMTP server to be used to email handling.</td>
</tr>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Use to modify the property name as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>The AIX Password (Use when overriding the value in the SymConfig Activity). As an alternative, if you wish to hide the password from viewing, create a User in JAMS, set the &quot;Logon As&quot; to the AIX Username, click the password button and enter the password for this account, and then use that JAMS UserName in the JAMSUsername workflow property.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX Server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>The Sym Number. Use when overriding the value in the SymConfig Activity. The parameter defaults to '000'.</td>
</tr>
<tr>
<td>UserName</td>
<td>The AIX UserName. Use when overriding the value in the SymConfig activity.</td>
</tr>
</tbody>
</table>

---

**FileToPC**
Downloads a SYM File from the FTP Server to the PC. The File is downloaded from the specified directory.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>The download location for the SYM File.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Used to modify the property name as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>The AIX Password (Use when overriding the value in the SymConfig Activity). As an alternative, if you wish to hide the password from viewing, create a User in JAMS, set the &quot;Logon As&quot; to the AIX Username, click the password button and enter the password for this account, and then use that JAMS UserName in the JAMSUsername workflow property.</td>
</tr>
</tbody>
</table>
### ServerDirectory
Server Directory beneath the '/SYMnnn/' folder

### ServerFileName
The Name of the file on SYM server in the specified Directory.

### ServerName
The Name of the AIX Server. Use when overriding the value in the SymConfig Activity.

### SymNumber
The SYM Number. Use when overriding the value in the SymConfig activity. Defaults to '000'

### UserName
The AIX Username. Use when overriding the value in the SymConfig activity.

---

**LetterFiletoPC**
Downloads a LetterFile from the FTP Server to the PC. The LetterFile is downloaded from the SYMnnn/LETTERSPECS directory, where nnn is a three-digit SYM Number (e.g., '007').

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>The local directory is the download location for the SYM LetterFile.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Used to modify the property name as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>Use when overriding the value in the SymConfig Activity. As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (4) enter the password for this account, and (5) then use that JAMS Username in the JAMSUserName workflow property.</td>
</tr>
<tr>
<td>ServerFileName</td>
<td>The name of the LetterFile on the SYM Server in the '/SYMnnn/LETTERSPECS' directory.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>Use when overriding the value in the SymConfig activity. This parameter defaults to '000'.</td>
</tr>
<tr>
<td>UserName</td>
<td>The AIX Username. Use when overriding the value in the SymConfig activity.</td>
</tr>
</tbody>
</table>

---

**PCToDataFile**
Uploads a DataFile from the PC to the FTP server. The DataFile is uploaded to the SYMnnn/DATAFILES directory, where nnn is a three-digit SYM Number (e.g., '007').

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>The local directory of the Data File to send to the '/SYMnnn/DATAFILES' directory on the SYM Server.</td>
</tr>
<tr>
<td>LocalFileName</td>
<td>The Local file name of the DataFile to upload to the '/SYMnnn/DATAFILES Directory on the SYM server.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Used to modify the property name as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>Use when overriding the value in the SymConfig Activity. As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (4) enter the password for this account, and (5) then use that JAMS Username in the JAMSUserName workflow property.</td>
</tr>
</tbody>
</table>
ServerName | The name of the AIX Server. (Use when overriding the value set in the SymConfig activity).
SymNumber | Use when overriding the value in the SymConfig Activity. This parameter defaults to '000'
UserName | The AIX UserName. Use when overriding the value in the SymConfig activity.

**PCToEditFile**
Uploads an EditFile from a local location to a remote folder via FTP operation

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overwrite</td>
<td>Turn overwriting on or off</td>
</tr>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>Local Directory of the Edit File to send to the '/SYMnnn/REPORT' directory on the server</td>
</tr>
<tr>
<td>LocalFileName</td>
<td>Local File Name of the Edit File to send to the '/SYMnnn/REPORT' directory on the server</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Used to modify the activity name property as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>Use when overriding the value in the SymConfig Activity. As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (4) enter the password for this account, and (5) then use that JAMS UserName in the JAMSUserName workflow property.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>The Sym Number. Use when overriding the value in the SymConfig activity. The parameter defaults to '000'.</td>
</tr>
<tr>
<td>UserName</td>
<td>The AIX UserName. Use when overriding the value in the SymConfig activity.</td>
</tr>
</tbody>
</table>

**PCToFile**
Uploads a file from the PC to the specified directory on the SYM server.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>The local directory of the file to send to the specified SYM directory</td>
</tr>
<tr>
<td>LocalFileName</td>
<td>The local File Name of the file to send to the specified SYM directory</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Used to modify the activity name property as displayed in the activity Panel.</td>
</tr>
<tr>
<td>Password</td>
<td>Use when overriding the value in the SymConfig Activity. As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (4) enter the password for this account, and (5) then use that JAMS UserName in the JAMSUserName workflow property.</td>
</tr>
<tr>
<td>ServerDirectory</td>
<td>Server directory beneath the '/SYMnnn/' folder where the file will be sent</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>The Sym Number. Use when overriding the value in the SymConfig activity. The parameter defaults to '000'.</td>
</tr>
</tbody>
</table>
**UserName**
The AIX UserName. Use when overriding the value in the SymConfig activity.

---

**PCToLetterFile**
Uploads a LetterFile from the PC to the FTP Server. The LetterFile is uploaded to the SYMnnn/LETTERSPECS directory, where nnn is a three-digit SYM Number (e.g., '007').

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>Local directory of the Letter File to send to the '/SYMnnn/LETTERSPECS' Directory on the Server</td>
</tr>
<tr>
<td>LocalFileName</td>
<td>Local File Name of the Letter File to send to the '/SYMnnn/LETTERSPECS' Directory on the Server</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Used to modify the activity name property as displayed in the activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>Use when overriding the value in the SymConfig Activity. As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (4) enter the password for this account, and (5) then use that JAMS Username in the JAMSUserName workflow property.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>The Sym Number. Use when overriding the value in the SymConfig activity. The parameter defaults to '000'.</td>
</tr>
<tr>
<td>UserName</td>
<td>The AIX UserName. Use when overriding the value in the SymConfig activity.</td>
</tr>
</tbody>
</table>

---

**ReportFileToPC**
Downloads a ReportFile from a remote location to a local folder via FTP

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX login. Any JAMS User.</td>
</tr>
<tr>
<td>LocalDirectory</td>
<td>The Local Directory where the Report File will be located.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Used to modify the property name as displayed in the Activity panel.</td>
</tr>
<tr>
<td>LocalFileName</td>
<td>The name for the local report file. The default is the source name.</td>
</tr>
<tr>
<td>Password</td>
<td>Use when overriding the value in the SymConfig Activity. As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (4) enter the password for this account, and (5) then use that JAMS Username in the JAMSUserName workflow property.</td>
</tr>
<tr>
<td>Report</td>
<td>The SymReport that identifies the report. This is returned by RunJobFile.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>The Sym Number. Use when overriding the value in the SymConfig activity. The parameter defaults to '000'.</td>
</tr>
<tr>
<td>UserName</td>
<td>The AIX UserName. Use when overriding the value in the SymConfig activity.</td>
</tr>
</tbody>
</table>

---

**Creating an in-memory copy of the desired report**

1. Click the "Variables" tab, located at the bottom of the Activity Panel, then select Create Variable
2. Name the variable the desired report name
3. Click on the listed Variable Type. A drop-down list will appear. Select Browse for Types from the dropdown list.
4. In the dialog that appears, type "SymReport" into the search box or navigate to the SymReport variable type. Select the SymReport variable type and click "OK".
5. Ensure the Scope is set to "Sequence".
6. Leave the "Default" column blank.
7. Select the ReportFileToPC activity in the Activity Panel, and set the "Report" property to the name of the created variable.

RunJobFile
Runs a JobFile on the Symitar System. By default JobFiles are run in synchronous mode which means that the program will wait for the JobFile to finish.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSSymUserID</td>
<td>For this JAMS user, enter the SymUserID as the &quot;logon as&quot; name, and set the password by clicking the &quot;Set Password&quot; button. When JAMS executes a job file, it will assemble the JAMSSymUserID as &quot;username.password&quot; (sans quotes).</td>
</tr>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>BatchSequenceNumber</td>
<td>Returns the batch sequence number of the Job.</td>
</tr>
<tr>
<td>JobFileBatchQueue</td>
<td>A numeric value (0-3) that will determine the batch queue to run the JobFile. Use 99 to find the first available empty queue. 0 will be used if 99 is specified and all queues are busy.</td>
</tr>
<tr>
<td>JobFileName</td>
<td>The name of the JobFile to run.</td>
</tr>
<tr>
<td>ReportsProduced</td>
<td>Returns a collection of reports produced by the job.</td>
</tr>
<tr>
<td>SymOperatorId</td>
<td>Operator ID to log into SYMOP. In most cases this will be &quot;0000&quot; or the same as the user's SymUserld.</td>
</tr>
<tr>
<td>SymUserId</td>
<td>SYM specific userid. Must be in the format of symuserid.password. As an alternative, if you want to hide your symuserid/password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the symuserid, (3) click the password button and enter the password for this symuserid, and then (4) use that JAMS User Name in the JAMSSymUserID workflow property. JAMS will then correctly format the symuserid and password.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Use to modify the property name as displayed in the Activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>The AIX Password (Use when overriding the value in the SymConfig Activity). As an alternative, if you wish to hide the password from viewing, (1) create a User in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and enter the password for this account, and then (4) use that JAMS User Name in the JAMSUsername workflow property.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX server. Use when overriding the value set in the SymConfig activity.</td>
</tr>
<tr>
<td>Sym</td>
<td>The Sym Number. Use when overriding the value in the SymConfig Activity. This parameter defaults to '000'.</td>
</tr>
<tr>
<td>TelnetPort</td>
<td>Enter the port number. This parameter defaults to '23'.</td>
</tr>
<tr>
<td>Username</td>
<td>The AIX UserName. Use when overriding the value in the SymConfig activity.</td>
</tr>
</tbody>
</table>

SymConfig
Creates a parent-child relationship for all SYM activities nested within SymConfig and set the default values for these child activities. If the child activities contain their own settings, those values will be used in place of the values set in the SymConfig activity.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMSUsername</td>
<td>The User ID for AIX Login. Any JAMS user.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Use to modify the property name as displayed in the Activity panel.</td>
</tr>
<tr>
<td>Password</td>
<td>The AIX password. As an alternative, if you wish to hide the password from viewing, (1) create a user in JAMS, (2) set the &quot;Logon As&quot; to the AIX Username, (3) click the password button and (3) enter the password for this account, and then (4) use that JAMS User Name in the JAMSUserName workflow property.</td>
</tr>
<tr>
<td>ServerFolderDirectory</td>
<td>The Path of the SYM Server File.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the AIX Server</td>
</tr>
<tr>
<td>ServerRunTimeoutInSecs</td>
<td>A numeric value (in seconds).</td>
</tr>
<tr>
<td>ServerTransferMode</td>
<td>ASCII or Binary.</td>
</tr>
<tr>
<td>SymNumber</td>
<td>The Sym Number.</td>
</tr>
<tr>
<td>UserName</td>
<td>The AIX UserName.</td>
</tr>
</tbody>
</table>

Still have questions about Symitar or Workflows? Try watching our [JAMS Workflow Video](https://example.com).  
NOTE: This video was recorded in JAMS V6. The sample Workflow principles shown there apply to JAMS V7.
z/OS Integration Pack

JAMS for z/OS incorporates z/OS Job scheduling into the JAMS platform and requires nothing to install on the z/OS host. When creating the agent used with the z/OS integration, you must select the z/OS agent type and the FTP platform.

Use this step-by-step guide to define a new z/OS Job using the JAMS Client.

Setting up the Connection with JAMS as a TSO User

To set up a connection with JAMS you will need to first define a TSO user on the mainframe as well as in JAMS (from the Credentials shortcut).

1. Open the Credential Definitions view from the JAMS Menu.
2. Click the Add button on the Control Bar.
3. The JAMS Credential definition dialog opens.
4. Set the Credential Name, Logon As, Password, and Password Confirmation values.
5. Click OK. By default, JAMS will open the full Credential Definition dialog after the Credential definition is initially saved.
6. On the Security tab of the Credential dialog, Add the ACE for the TSO user.
   Please note: this user must have login rights as well as rights to execute items that will be added in later JAMS jobs.
7. Save and Close the Credential definition.


JAMS interfaces with JES as a TSO user. The TSO login ID is interpreted by JAMS as a valid JAMS User, which is encrypted on the JAMS side and controlled by RACF on the z/OS side. The TSO user must have permissions to submit and receive Jobs on both platforms and the TSO username must contain 7 characters or less.

On the JAMS side, the z/OS Integration must be installed on the JAMS Server.

Setting up a z/OS Job in JAMS

1. To create a z/OS Job in JAMS
2. Select the desired folder for the z/OS Job, then ensure the Job Definitions tab is selected.
3. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
4. In the dialog, give the new Job a Name, Description (optional), and Execution Method.
   In this case, select z/OS to create a z/OS Job.
5. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
6. The Job Definition dialog will open.
   NOTE: An Execute As property should be configured with the previously defined TSO user.
   NOTE: An Agent property should be configured with the Mainframe Agent hostname or IP address.
8. On the Source tab, insert the JCL code.
9. Save and Close the Job definition.

How does JAMS run a Mainframe Job?

Jobs are stored on an FTP server located on the mainframe using a proprietary IBM-based FTP language. JAMS communicates with the mainframe using this FTP language. Once a Job is executed in JAMS an FTP connection is opened to the FTP Daemon in JES and the Job is sent to the mainframe and executed.

JAMS monitors the Job spool output and waits for the Job to complete by polling the FTP connection. Once the output files are complete JAMS purges the files from the JES spool.

Steps can be run in JES in the same way as tasks in a JAMS Sequence. JCL defines the behavior of the Job depending on what is happening with a particular Job task.

What Information does JES Return?

The JAMS Scheduler retrieves all Job related output from the JES Spool. This includes any DD statements with SYSOUT held on the spool. The output appears in the JAMS log and can be used by operations administrators to
debug problems and, if necessary, restart the Job.

JAMS for z/OS parses the Job output and provides job success (“0”) or failure information for the JAMS Scheduler to react and report on a variety of message codes, including HASP and return code analysis.

**What happens when a Job fails?**

If a Job fails due to a network outage or other problem all relevant information is included in the output and also displayed in the JAMS log. Customization of these messages can be set using a configuration file.

A special option in the Job definition includes a listing of Jobs on the JES Spool allowing a Job to rerun on the JAMS side for output recovery.

**JAMS for z/OS Capabilities**

**Manages and Submits JCL Files to JES2 on z/OS for Execution**

JAMS provides the ability to store a JCL source in its internal database. The JCL then becomes a part of the JAMS Job Definition along with other parameters and information needed to run the job.

**Schedules Jobs with Dependencies on Multiple z/OS Hosts**

The JAMS Scheduler can support multiple z/OS hosts and track Job dependencies between them and non-z/OS hosts.

**The JAMS Scheduler Retrieves and Displays the Job Output**

The JAMS Scheduler retrieves all job related output from the JES Spool. This includes any DD statements with SYSOUT held on the spool. The output appears in the JAMS log and can be used by operations administrators to debug problems and, if necessary, restart the jobs.

**Includes Automatic Determination for all Job Results**

JAMS for z/OS parses the job output and provides job success or failure information for the JAMS Scheduler to react and report on a variety of message codes, including HASP and return code analysis.

**Utilizes Standard JES Settings**

JAMS for z/OS uses JESINTERFACELEVEL 1 and doesn’t require special customization of Parmlib settings.

**Requires No installation on z/OS Hosts**

JAMS for z/OS uses z/OS FTP with JES and requires no additional mainframe software installation.

**Provides Automatic Job Purging Capabilities**

JAMS for z/OS automatically purges the job output files for the JES2 Spool after each Job has completed and been retrieved.

**Provides Reliable Tracking of Jobs Running on the Mainframe**

The JAMS Scheduler can recover its connection to JES if it is lost due to a network outage or other problem, even when a job is in the middle of executing on z/OS. In the event of failure on the JAMS server, JAMS can restart for that specific job and seamlessly recover.

**Uses a Secure TSO User Login**

JAMS interfaces with JES as a TSO user. The TSO login ID is interpreted by JAMS as a valid JAMS user and encrypted on the JAMS side and controlled by RACF on the z/OS side.

**Provides Full Integration with the JAMS Scheduler**

JAMS for z/OS becomes part of the integrated capabilities of the JAMS Scheduler across different platforms allowing it to control jobs based on dependencies on all supported hosts.
JAMS for z/OS is Easy to Set Up

Setting up a z/OS job is now as easy as defining any other JAMS Job.

Provides a Listing of Pending Jobs on the z/OS Spool

A special option in the JOB definition includes a listing of JOBS on the JES Spool allowing a job to rerun on the JAMS side for output recovery.

Scenario: Synchronizing Data with JAMS for z/OS

Suppose an organization has the need to reliably synchronize data from an OLTP system based on a MS SQL Server or Oracle with a mainframe-based database. This scenario is easily implemented with JAMS for z/OS.

In order to accomplish this the JAMS Scheduler would initially run a job, based on an event, that generates files comprised of data extracted from a MS SQL Server database. It would then upload those files to the z/OS platform via FTP before submitting JCL through JAMS for z/OS. These uploaded files would then update a mainframe database. The reverse scenario is just as easy where a MS SQL Server is updated with files generated by the mainframe.
Oracle EBS Integration
Creating Oracle EBS Jobs

JAMS natively supports automation using many leading business applications, including the Oracle E-Business Suite. Integration Packs for this and other preferred applications are selected during the JAMS installation process. Once the Oracle E-Business Suite Integration is enabled, users can create, manage, deploy and monitor every kind of Oracle EBS batch process job, regardless of where it resides.

In This Topic:
- Installing the Oracle EBS Integration Pack
- Configuring Oracle EBS Connection Strings
- Creating Oracle EBS Jobs
  - Creating OracleEBSConcurrent Jobs
  - Creating OracleEBSJobSsh Jobs
  - Creating OracleEBSRequestSet Jobs

Installing the Oracle E-Business Suite Integration Pack

To work with Oracle EBS in JAMS, users must install the Oracle E-Business Suite Integration Pack on the JAMS Scheduler Server and deploy an Agent to the Oracle Server. For more information on deploying Agents, see Agent D and Agent X (JAMS AgentX) in the on-line documentation sections.

- On the JAMS Scheduler Server, run the JAMS Installer.
- On the Feature Selection step, ensure the JAMS Integration Pack for Oracle E-Business Suite component is checked. Do not un-check any components unless those components should be uninstalled.
- Finish the installation.

NOTE: Un-checking components in the Feature Selection step will uninstall those components.

Configuring Oracle EBS Connection Strings

The Oracle EBS Connection String allows users to connect to Oracle from within JAMS. This connection string must be properly configured to run Oracle Jobs in JAMS. The Oracle EBS connection should be configured on each Oracle EBS Execution Method in JAMS.

Users may override the connection string configured on an Execution Method by configuring a Connection String Parameter at the Folder level or Job level.

NOTE: If the user on an Oracle EBS Job has OS Authentication permissions to Oracle, then JAMS will pass the credentials from the User into the defined connection string during execution, and an Oracle user is not needed in the Job's source. If the User on the Job does not have OSI Authentication, an Oracle user must be defined in the dropdown list that has permissions to the Oracle Database.

Configure the Oracle EBS Connection on the Execution Method

1. Run the JAMS Client as an administrator.
2. Select the Execution Methods shortcut from the Menu.
3. Double click on the given Oracle EBS Execution Method to open the Execution Method Properties dialog.
4. On the Parameters tab, select the OEBSConnectionString Parameter, then click Properties to open the Parameter Properties dialog.
5. Set the Default Value to the following, with "Host=Orchid64" and "SERVICE_NAME=MVP" adjusted to values that match the given environment.

```
User Id = {0}; Password = {1}; Data Source = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = Orchid64)(PORT = 1521))(CONNECT_DATA = (SERVICE_NAME = MVP)));
```

7. Save and Close the Execution Method dialog.

Configure the Oracle EBS Connection on the Folder level

A Connection String should ONLY be defined at the Folder level if users wish to override the Execution Method level Connection Strings. Note that the Folder level Connection String will apply to all Oracle EBS Jobs within that folder.

1. Right click on the desired Folder and select Properties from the drop-down list to open the Folder Properties dialog.
2. Select the Parameters tab, and check for a Parameter named OEBSConnectionString. If this Parameter does not yet exist on the folder, click the Add button.
3. In the Add a Parameter wizard, name the Parameter OEBSConnectionString.
4. Ensure the Data Type is set to Text.
5. Set the Default Value to the following, with "Host=Orchid64" and "SERVICE_NAME=MVP" adjusted to values that match the given environment.

```
User Id = {0}; Password = {1}; Data Source = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = Orchid64)(PORT = 1521))(CONNECT_DATA = (SERVICE_NAME = MVP)));
```

6. Save and Close the Folder Properties.

Configure the Oracle EBS Connection on the Job level

A Connection String Parameter should ONLY be defined at the Job level if users wish to override Execution Method or Folder level Connection Strings.

1. Right click on a Job and select Properties from the drop-down list to open the Job Properties dialog.
2. Select the Parameters tab, and check for a Parameter named OEBSConnectionString. If this Parameter does not yet exist on the Job, click the Add button.
3. In the Add a Parameter wizard, name the Parameter OEBSConnectionString.
4. Ensure the Data Type is set to Text.
5. On the Parameter Value tab, set the Default Value to the following, with "Host=Orchid64" and "SERVICE_NAME=MVP" adjusted to values that match the given environment.

```
User Id = {0}; Password = {1}; Data Source = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = Orchid64)(PORT = 1521))(CONNECT_DATA = (SERVICE_NAME = MVP)));
```


Creating Oracle EBS Jobs in JAMS

JAMS offers OracleEBSConcurrent, OracleEBSJobSsh, and OracleEBSRequestSet Execution Methods to run Oracle EBS Jobs.

The OracleEBSConcurrent Execution Method gives users a series of automatically populated drop-down fields to run Concurrent Jobs, including the ability to view and set each parameter on the Concurrent Job from within JAMS.

The OracleEBSJobSsh Execution Method gives users a fill in the blank style form to run Concurrent Jobs from within JAMS. The advantage of the fill in the blank form over the OracleEBSConcurrent Execution Method is that field values in the Job Source may be defined as JAMS Parameters. By using Parameters in the Job Source, users can create a self-service Oracle Job from within JAMS.
Creating OracleEBSConcurrent Jobs in JAMS

1. Select the desired folder for the OEBSCOncurrent Job, then ensure the Job Definitions tab is selected.
2. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a Name, Description (optional), and Execution Method.
   In this case, select OracleEBSConcurrent to create an Oracle EBS Concurrent Job.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
5. The Job Definition dialog will open.
7. On the Source tab, set the Oracle User. This user should have access to the Oracle Database.
8. Select the Application User from the automatically populated drop-down list.
9. Select the Responsibility group from the automatically populated drop-down list.
10. Select the relevant Application from the automatically populated drop-down list.
11. Select the Concurrent Program to be run from the automatically populated drop-down list.
12. The Parameters section of the Source will automatically populate with the Concurrent Program's available Parameters.
13. Set Source Parameters as desired, ensuring that required Parameters have values.
14. Click Save and Close to finish editing the Job definition.

Creating OracleEBSJobSsh Jobs in JAMS

OracleEBSJobSsh Jobs in JAMS provide a fill-in-the-blank style form for users to configure their Concurrent Oracle Jobs in JAMS. The benefit of the fill-in-the-blank form over the automatically populated drop-down list of the OracleEBSConcurrent Execution Method is the parametrization of field values. By using Parameters in the Job Source, users can create a self-service Oracle Job from within JAMS.

Create an OracleEBSJobSsh Job in JAMS:

1. Select the desired folder for the OracleEBSJobSsh Job, then ensure the Job Definitions tab is selected.
2. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a Name, Description (optional), and Execution Method.
   In this case, select OracleEBSJobSsh to create an Oracle EBS Ssh Job.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
5. The Job Definition dialog will open.
7. In the Source, specify the information on the Job to run in Oracle EBS. See the available fields and their Oracle counterparts below.

<table>
<thead>
<tr>
<th>JAMS</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Name</td>
<td>Program Shortname</td>
</tr>
<tr>
<td>Application Short Name</td>
<td>Program Application Shortname</td>
</tr>
<tr>
<td>Responsibility Short name</td>
<td>Responsibility App Shortname</td>
</tr>
<tr>
<td>Responsibility Name</td>
<td>Responsibility Name</td>
</tr>
<tr>
<td>Username</td>
<td>Username</td>
</tr>
<tr>
<td>Parameters</td>
<td>Parameters</td>
</tr>
</tbody>
</table>

Note that Parameters in the Parameters field are space delimited.
8. Save and Close the Job definition.

Creating OracleEBSRequestSet Jobs in JAMS

1. Select the desired folder for the OracleEBSRequestSet Job, then ensure the Job Definitions tab is selected.
2. Click the Add button from the Control Bar to open the Add a New JAMS Job Definition dialog.
3. In the dialog, give the new Job a Name, Description (optional), and Execution Method.
   In this case, select OracleEBSRequestSet to create an Oracle EBS Request Set Job.
4. By default, the full Job Definition dialog will open when the Job is initially saved. Click OK.
5. The Job Definition dialog will open.
7. On the Source tab, set the Oracle User. This user should have access to the Oracle Database.
8. Select the Application User from the automatically populated drop-down list.
9. Select the Responsibility group from the automatically populated drop-down list.
10. Select the relevant Application from the automatically populated drop-down list.
11. Select the Request Set to be run from the automatically populated drop-down list.
12. The Request Set Job section will populate with a list of the Concurrent Programs in the Request Set.
13. Select a Concurrent Program and then click the Parameters button to open the parameters of a Concurrent Program within the Request Set.
14. In the Parameters dialog, set Parameter values as desired, then click OK.
15. Click Save and Close to finish editing the Job.
Configuring The OracleStoredProc Execution Method

Configuring the OracleStoredProc Execution Method:

In order to use the Oracle Stored Procedure Execution Method you will first need to have a user name, password and a datasource. As long as the datasource is in a format that is compatible with an Oracle Connection String, it will open a connection to the Oracle DB server.

The Oracle Stored Procedure relies on a default value being present in the parameters of the Execution Method, which can be overridden at the Folder or Job level.

The default format for the Execution Method is:

- User Id={0}
- Password={1}
- DataSource={2}

NOTE: By default, the User Id, Password, and DataSource set on the given OracleStoredProc Job will be passed into the "{}" braces on the Execution Method. Users do not need to edit the OracleConnectionString parameter by default.

If using a different connection string, you will need to override the original connection string by creating a new Parameter at the Folder or Job level called OracleConnectionString.

Once the ConnectionString format has been set, include a datasource and User Name in the Execution Method’s Source tab. Establish the connection. Once the User has authorized access to the Oracle DB Server, click on the dropdown list to view a populated listing of stored procedures.

NOTE: Users without a Data Source Name (DSN) for their Oracle Connection will need to specify the entire traditional Oracle Connection String in the Data Source field of the OracleStoredProc Job’s Source tab.

When a stored procedure is selected, the executable will display the parameter values associated with the stored procedure and prepend a “@” symbol in order to distinguish the procedure parameters from the executable parameters.

Once all the properties have been populated, save and submit the Job.

Note: Submit Options located in the Job properties can now be used to manually enter your parameter values.