Phoenix® Job Management System™ User's Guide

Applies to:

Phoenix WinNonlin® 8.3
Phoenix NLME™ 8.3
IVIVC Toolkit™ 8.3 for Phoenix
AutoPilot Toolkit™ 8.3 for Phoenix

Legal Notice

Phoenix® WinNonlin®, Phoenix NLME™, IVIVC Toolkit™, CDISC® Navigator, Certara Integral™, PK Submit™, AutoPilot Toolkit™, Job Management System™ (JMS™), Trial Simulator™, Validation Suite™ copyright ©2005-2020, Certara USA, Inc. All rights reserved. This software and the accompanying documentation are owned by Certara USA, Inc. The software and the accompanying documentation may be used only as authorized in the license agreement controlling such use. No part of this software or the accompanying documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, except as expressly provided by the license agreement or with the prior written permission of Certara USA, Inc.

This product may contain the following software that is provided to Certara USA, Inc. under license: ActiveX® 2.0.0.45 Copyright © 1996-2020, GrapeCity, Inc. AngleSharp 0.9.9 Copyright © 2013-2020 AngleSharp. All rights reserved. Autofac 4.8.1 Copyright © 2014 Autofac Project. All rights reserved. Crc32.Net 1.2.0.5 Copyright © 2016 force. All rights reserved. Formula One® Copyright © 1993-2020 Open-Text Corporation. All rights reserved. Json.Net 7.0.1.18622 Copyright © 2007 James Newton-King. All rights reserved. LAPACK Copyright © 1992-2013 The University of Tennessee and The University of Tennessee Research Foundation; Copyright © 2000-2013 The University of California Berkeley; Copyright © 2006-2013 The University of Colorado Denver. All rights reserved. Microsoft® .NET Framework Copyright 2020 Microsoft Corporation. All rights reserved. Microsoft XML Parser version 3.0 Copyright 1998-2020 Microsoft Corporation. All rights reserved. MPICH2 1.4.1 Copyright © 2002 University of Chicago. All rights reserved. Minimal Gnu for Windows (MinGW, http://mingw.org/) Copyright © 2004-2020 Free Software Foundation, Inc. NLog Copyright © 2004-2020 Jaroslaw Kowalski <jaak@jkowalski.net>. All rights reserved. Reinforced.Typings 1.0.0 Copyright © 2020 Reinforced Opensource Products Family and Pavel B. Novikov personally. All rights reserved. RtfToHtml.Net 3.0.2.1 Copyright © 2004-2017, SautinSoft. All rights reserved. Sentinel RMS™ 8.4.0.900 Copyright © 2006-2020 Gemalto NV. All rights reserved. Syncfusion® Essential Studio for WinForms 16.4460.0.42 Copyright © 2001-2020 Syncfusion Inc. All rights reserved. TX Text Control .NET for Windows Forms 26.0 Copyright © 19991-2020 Text Control, LLC. All rights reserved. Websites Screenshot DLL 1.6 Copyright © 2008-2020 WebsitesScreenshot.com. All rights reserved. This product may also contain the following royalty free software: CsvHelper 2.16.3.0 Copyright © 2009-2020 Josh Close. DotNetbar 1.0.0.19796 (with custom code changes) Copyright © 1996-2020 DevComponents LLC. All rights reserved. ImageMagick® 5.0.0.0 Copyright © 1999-2020 ImageMagick Studio LLC. All rights reserved. IMSL® Copyright © 2019-2020 Rogue Wave Software, Inc. All rights reserved. Ninject 3.2 Copyright © 2007-2012 Enkari, Ltd. Software for Locally-Weighted Regression Authored by Cleveland, Grosse, and Shyu. Copyright © 1989, 1992 AT&T. All rights reserved. SQLite (https://www.sqlite.org/copyright.html). Ssh.Net 2016.0.0 by Olegkap Drieseng. Xceed® Zip Library 6.4.17456.10150 Copyright © 1994-2020 Xceed Software Inc. All rights reserved.

Information in the documentation is subject to change without notice and does not represent a commitment on the part of Certara USA, Inc. The documentation contains information proprietary to Certara USA, Inc. and is for use by its affiliates' and designates' customers only. Use of the information contained in the documentation for any purpose other than that for which it is intended is not authorized. NONE OF CERTARA USA, INC., NOR ANY OF THE CONTRIBUTORS TO THIS DOCUMENT MAKES ANY REPRESENTATION OR WARRANTY, NOR SHALL ANY WARRANTY BE IMPLIED, AS TO THE COMPLETENESS, ACCURACY, OR USEFULNESS OF THE INFORMATION CONTAINED IN THIS DOCUMENT, NOR DO THEY ASSUME ANY RESPONSIBILITY FOR LIABILITY OR DAMAGE OF ANY KIND WHICH MAY RESULT FROM THE USE OF SUCH INFORMATION.

Destination Control Statement

All technical data contained in the documentation are subject to the export control laws of the United States of America. Disclosure to nationals of other countries may violate such laws. It is the reader's responsibility to determine the applicable regulations and to comply with them.

United States Government Rights

This software and accompanying documentation constitute "commercial computer software" and "commercial computer software documentation" as such terms are used in 48 CFR 12.212 (Sept. 1995). United States Government end users acquire the Software under the following terms: (i) for acquisition by or on behalf of civilian agencies, consistent with the policy set forth in 48 CFR 12.212 (Sept. 1995); or (ii) for acquisition by or on behalf of units of the Department of Defense, consistent with the policies set forth in 48 CFR 227.7202-1 (June 1995) and 227.7202-3 (June 1995). The manufacturer is Certara USA, Inc., 100 Overlook Center, Suite 101, Princeton, New Jersey, 08540.

Contents

Job Management System (JMS)

The Phoenix Job Management System (JMS) provides features to facilitate remote execution of work-flows. These features allow you to: submit long running processes without tying up your desktop, shut down your desktop or Phoenix while the processes are running remotely, and reclaim the results at a future time.

The following topics are discussed in this section:

Installing JMS

During the installation of Phoenix the user will have access to installation of the JMS components via selection of either a Complete installation or Custom installation type.
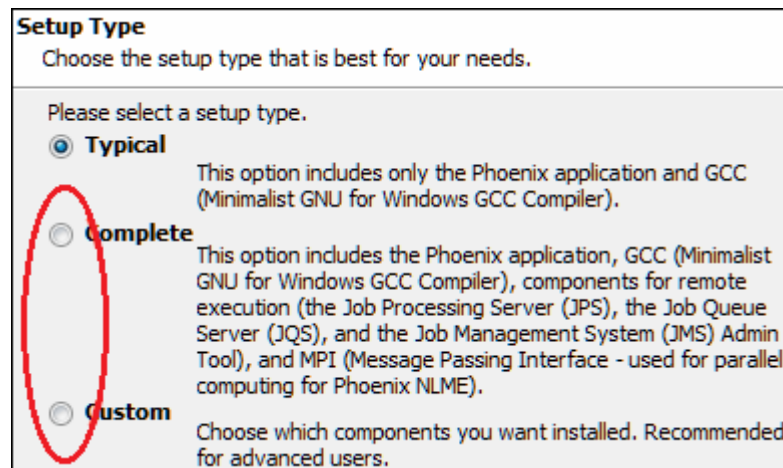

Figure 2-1. Setup page

Complete installation

1. Select the **Complete** option to install all components (MPI: Message Passing Interface and JMS servers: Job Queue Server and Job Processing Server).

2. Click **Next >** and, if desired, change the installation directory.
   No further selections are needed to setup JMS in the remainder of the installation.

3. Click **Next >** to review the installation settings.

4. Click **Next >** to install the components.

5. When the installation is complete, use the **Restart now** checkbox to indicate whether or not to reboot the computer at this time or not and click **Finish**.

Note: To complete the installation, the system on which the JMS component(s) are installed must be rebooted.

After rebooting, the installed Server parts will start as a Windows Service. Below is a representation of both JMS Servers installed on a single PC.
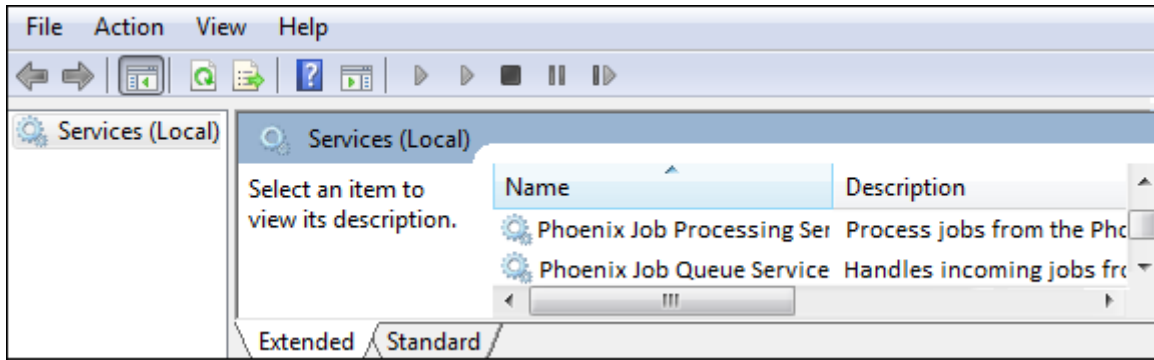

Figure 2-2. Windows Services dialog listing both JMS servers

Note: For third-party applications compatible with Phoenix to be utilized for Remote Execution, the Job Processing Service must be run as an existing user on the JPS system, and not just the default Local System account. The supported third-party applications are R, SAS, and NONMEM, which must be installed and have working folders on the JPS and configured via the Phoenix Preferences on this system. The deployed Phoenix installation on the JPS has no need to be licensed as its sole purpose is to configure the third-party applications for use by the JPS.

Custom installation

1. Select the **Custom** installation type to choose individual components for installation.

    This is useful in an environment where components will reside on multiple computers, for example Job Queue Server on one server and the Job Processing Server on another server. This configuration will facilitate distributing the load of processing for the Queue and the Engine(s) for processing.

2. Click **Next >** to display the Select Features page listing the optional components available for installation.
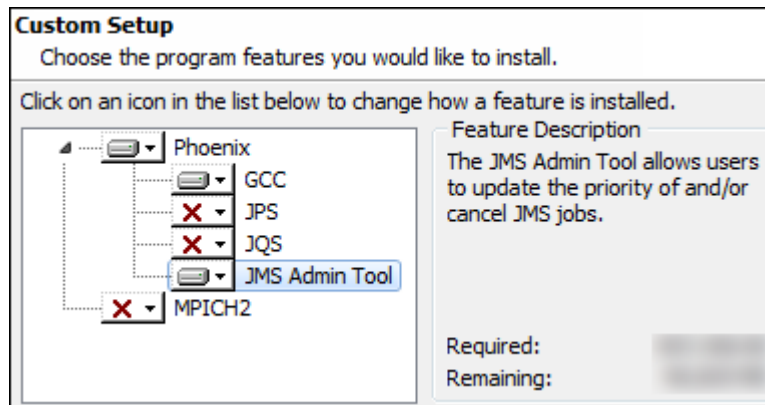

Figure 2-3. Select Features page

3. Select a component using its drop-down menu and choose from the available options.
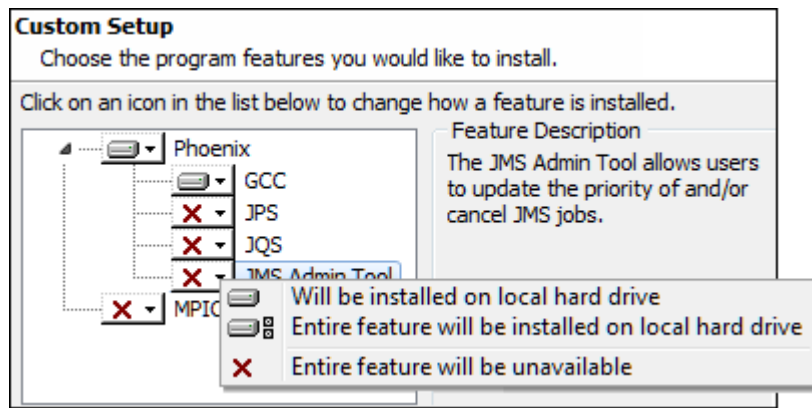
Figure 2-4. Custom Setup page

It is preferred to select the option **Entire feature will be installed on local hard drive**.

4. Click **Next >** and, if desired, change the installation directory.

5. Click **Next >** to review the installation settings.

6. Click **Next >** to install the components.

7. When the installation is complete, use the **Restart now** checkbox to indicate whether or not to reboot the computer at this time or not and click **Finish**.

Note: To complete the installation, the system on which the JMS component(s) are installed must be rebooted.

Configuring JMS

Instructions for configuring JMS follow. For more information on the configuration files mentioned, see "Descriptions of configuration files in Phoenix".

To configure the jqs.exe.config to set the processing options

1. Open `<Phoenix_install_dir>\application\jqs.exe.config`.

2. Go to the `jobqueue` and configure the following options.

   `isLocalJobQueue`: Leave as `false`.

   `maxHistory`: Set the maximum number of finished/canceled jobs to keep on the queue. When this number is exceeded the oldest jobs will be removed from the queue.

   `jobTimeoutInSeconds`: Set the amount of time in seconds that the queue waits to determine if a JPS machine has gone down while processing a job. If the queue does not get a heartbeat back from the processing machine continuously within this time frame, the job will be put back on the queue. Any results from the machine that timed out will not be accepted.

To configure the jobManagement.xml file to point to the correct JQS

1. Stop JPS.

2. Using a text editor, open `<Phoenix_install_dir>\application\jobManagement.xml`.

3

3. Modify the `machineName` attribute to point to the JQS machine. This value can be an IP address or a resolvable hostname.

4. In the `channels` section, modify the `port` attribute for HTTP and/or TCP.

5. In the `server` section, set `default=true` for the channel you wish to use (HTTP or TCP).

6. Restart the server.

Console mode can be used to verify that the service is setup properly and processing jobs.

To configure jps.exe.config to set the processing options

1. Open `<Phoenix_install_dir>\application\jps.exe.config`.

2. Go the `jobProcessingSection` and configure the following options:

   `maxProcessingThreads`: Set to the maximum number of concurrent JPS jobs to process on the machine. Factors to consider when setting the value include the number of available cores and whether or not MPI is also being used.

   `useLocalConfigurationFile`: Leave as `true`.

   `configurationFilePath`: Leave as empty.

   `threadPollInterval`: Set how often in seconds the JPS will ask the queue for a job if it is not already processing one.

   `gridAvailable`: Leave as `false`.

   `heartbeatTimeInSeconds`: Set how often in seconds JPS will talk back to the queue to report that it is still processing a job. This is used in conjunction with the time out option in the queue to determine if a processing machine has gone down and the job should be returned to the queue for processing.

3. Configure the HTTP/TCP channels to pick the port that will be used to communicate with Phoenix clients and JPS nodes.

4. To have JPS use a named user license server, remove the comment "`<!--`" before and after the `<appSettings>` section. Change the default IP address to the named user license server's IP address.

   `<add key="floatLicenseServer" value="127.0.0.1"/>`

   The Job Processing Server Sandbox directory is defined by `jps.exe.config`. The default location is `C:\Phoenix.JPS`. Some users might need to define another location if the default location is not acceptable. The directory name must be `\Phoenix.JPS` for the system to correctly function.

5. Users can change the sandbox directory drive letter to the letter of any local drive. The sandbox cannot be located on a networked hard drive.

   The sandbox location is defined in the following two areas in `jps.exe.config`:

   ```
   jobSandboxDirectory="C:\Phoenix.JPS"
   repositoryType="Certara.Phoenix.Services.\
       ObjectBrowser.FlatRepository.FlatRepository,\
       Services.ObjectBrowser"
   ```

```
repositoryDirectory="c:\Phoenix.JPS"
repositoryFormatType="Binary"
```

To configure the JPS to use the RPS

The other part of the process is to configure the JPS machine to send NONMEM executions to the RPS Queue.

1. Stop the JPS.

2. Using a text editor, open `PhoenixRemoteSettings.xml` which is located in the Phoenix application folder on the JPS machine.

3. Modify the server element so that it references the RQS server.

4. Set `executeRemotely` to `true` for all process types you wish to be executed by the RPS. All other process types would be `false`.

5. After saving your changes to `PhoenixRemoteSettings.xml`, restart JPS.

Descriptions of configuration files in Phoenix

Phoenix has various configuration files that are used to configure the Job Management System and set default configurations for the Phoenix desktop application. The configuration files are for the servers that process jobs submitted for remote execution. The content of these files are described in the following sections.

Desktop application and job processing servers

The Phoenix desktop application stores its settings in binary files that can only be modified in the Phoenix user interface. These files are stored in the users' `%APPDATA%\Certara\Phoenix` directory. For example, `C:\Users\<user name>\AppData\Roaming\Certara\Phoenix\Configuration`. The files can be moved to other computers and are only computer-specific in the areas of configuration files related to mapped drives. The items that can be configured are Autosave, SAS, NONMEM, R, Certara Integral, and Licensing. These settings are configurable in the Phoenix user interface by selecting **Preferences** in the **Edit** menu. This displays a dialog with configurable items. Once the settings are modified, the files can be redistributed to other computers.

Job Management System

The Job Management System consists of the client (Phoenix desktop application) that submits jobs for remote execution, the Job Queue where jobs are stored for processing, and the Job Servers, which processes the jobs in the queue.

The client configuration file `jobManagement.xml` is used to point the client to the Job Queue. This file allows users to configure the Job Queue location and the port number to use. This file is located in the same directory as the application data configuration files mentioned earlier.

The Job Queue configuration file `jqs.exe.config` is used to set the ports to listen to and except for the license location is generally not modified.

The Job Server configuration file `jps.exe.config` is used to define the connection between the Job Server and the Job Queue. It has other settings that typically do not need modification.

All these `*.config` files are in the application root directory. For example, `%INSTALL_-FOLDER%\Phoenix\application`.

Execution of AutoPilot Toolkit objects using JMS

Phoenix projects that contain AutoPilot Toolkit objects can only be executed on JMS using the 32-bit versions of the executables named `jqs32.exe` and `jps32.exe`. The configuration files for these executable files are `jqs32.exe.config` and `jps32.exe.config`. The same instructions apply to editing of the configuration files as outlined above. The installed Windows Services for the Job Queue Server and Job Processing Server will need to be set to manual or disabled. Console mode is necessary for the execution of AutoPilot Toolkit objects on JMS. This is due to the fact that the third party tool SigmaPlot can only be used by JMS when running in Console mode. Console mode does not affect the ability to execute any of the supported Third Party Tools.

Phoenix configuration file

Name: `Phoenix.exe.config`

Format: XML

Configurable Sections: Application Settings and Logging (uses NLog: see http://nlog-project.org more information).

```
<appSettings>
      <add key="KEY" value="VALUE"/>
</appSettings>
```

Keys and Values:

`hideCommute=true/false`: Toggles the ability for the user to check out commuter licenses.
`useDebugDirectory`: Location for NLME Debug files.

```
<nlog>
      <…>
<\nlog>
```

Configurations:

`<nlog globalThreshold="Trace">`: Defines the logging level, default is `Trace`. Valid values include: `Trace, Debug, Info, Warn, Error, Fatal`

JPS and JQS configuration files

Names: `jps.exe.config` and `jqs.exe.config`

Format: XML

Configurable Sections: Application Settings and Logging (uses NLog: see http://nlog-project.org more information).

```
<appSettings>
      <add key="KEY" value="VALUE"/>
</appSettings>
```

Keys and Values:

`floatLicenseServer`: States where the JPS named user license server is via IP Address.
`serviceLicense`: JPS/JQS license
`useDebugDirectory`: Location for NLME Debug files.

```
<nlog>
        <…>
<\nlog>
```

Configurations:

`<nlog globalThreshold="Trace">`: Defines the logging level, default is `Trace`. Valid values include: `Trace, Debug, Info, Warn, Error, Fatal`

`jqs.exe.config` only:

```
<jobqueue isLocalJobQueue="false" maxHistory="25"
    jobTimeoutInSeconds="300">
    <channels>
        <channel type="tcp" port="4000" />
        <channel type="http" port="8000" />
    </channels>
</jobqueue>
```

`jps.exe.config` only:

```
<jobProcessing maxProcessingThreads="1"
    useLocalConfigurationFile="true"
    configurationFilePath="" threadPollInterval="60"
    gridAvailable="false" heartbeatTimeInSeconds="60" />
```

`threadPollInterval`: How often to look for a job.
`heartbeatTimeInSeconds`: How often to report back to the queue when running the job.
`maxProcessingThreads=Ignore/Do Not Modify`
`gridAvailable=Ignore/Do Not Modify`
`maxProcessingThreads=Ignore/Do Not Modify`
`useLocalConfigurationFile=Ignore/Do Not Modify`
`configurationFilePath=Ignore/Do Not Modify`

Job management configuration file

Name: `jobManagement.xml`

Format: XML

Configurable Sections: Users can only modify the machine name and ports.

Example:

```
<jobManagement defaultPollInterval="60"
    usePolling="true">
  <channels>
    <channel name="tcp4000" type="tcp" port="4000"
        direction="out" />
    <channel name="http8000" type="http" port="8000"
        direction="out" />
  </channels>
  <servers localJobQueueChannelName="tcp4000">
    <server machineName="localhost" primary="true">
      <serverChannel channelName="tcp4000"
          default="true" />
      <serverChannel channelName="http8000" />
    </server>
```

```
        </servers>
    </jobManagement>
```

Plug-in configuration files

Location: `%APPDATA%\Certara\Phoenix`

Summary: Binary files that contain the settings found in the *Preferences* dialog (**Edit > Preferences**). They cannot be modified manually. They can only be modified by running an instance of Phoenix on the computer. After they are set by Phoenix, they can be distributed to other computers that have Phoenix installed.

Plug-In files:

> `NONMEM.binconfig`: NONMEM configuration information.
> `SAS.binconfig`: SAS configuration information.
> `pks_config.binconfig`: Certara Integral instance information.
> `pks_saveoptions.binconfig`: Certara Integral default save options.
> `ObjectBrowser.binconfig`: General project settings and the default save options.

If licensing configuration files are moved to another computer, the following three files must be moved together:

> `licensing.licenses.configuration.binconfig`
> `licensing.server.configuration.binconfig`
> `licensing.service.configuration.binconfig`

Configuring JMS to work with RPS

In addition to being able to execute Phoenix objects, the JPS can be configured to send certain pieces of work to a remote server for execution. For instance, assume that NONMEM is already installed on a server somewhere in the organization. The JPS can be configured so that all NONMEM jobs are executed on the server where NONMEM and the Phoenix Remote Server are already installed. Phoenix objects that can be setup for remote execution include Phoenix Model, NONMEM Shell, R Shell, SAS Shell, and PsN Shell.

When JPS sends part of a job for remote execution, JPS will wait for the execution results to return before continuing with the rest of the job.

Topics in this section include:

- PhoenixRemoteSettings.xml file

- Deploying the RPS Queue and Processor

PhoenixRemoteSettings.xml file

Configurable settings for remote execution are stored in the `PhoenixRemoteSettings.xml` file. The application will first look for this file in `\<user_data>\Application Data\Certara\Phoenix`. This is where the file will be saved if it has been updated using **Edit > Preferences > Remote Settings**. If the file is not found there, the copy located in `<Phoenix_install_directory>\Phoenix\application` will be used.

To set up log file details

1. In the `PhoenixRemoteSettings.xml` file, look for the following line:

   ```
   <settings logLevel="prod" logPath="" enabled="false">
   ```

2. Set the amount of information to store in the log file (`logLevel`). Levels include:

   `prod`: Production is the default level and the least verbose.
   `test`: Testing reports slightly more detail than `prod`.
   `debug`: Essentially a dump of information.

3. Set the location in which to store the log files generated by the Remote Phoenix Server (`log-Path`). If `logPath` is not specified, log files are written to `\<user_data\Application Data\Certara\Phoenix\Logs`.

To set up the remote execution server

1. In the `PhoenixRemoteSettings.xml` file, look for the following line:

   ```
   <server name="localhost" root="PhoenixServer" port="8080"
       useSSL="false" timeout="300" />
   ```

2. Edit the values to reference the RQS server.

   `name`: Server name
   `root`: Context root
   `port`: Server port
   `use SSL`: `true/false` Whether or not to use the secure socket layer.
   `timeout`: Server timeout value.

To set up remote execution of an object

1. In the `PhoenixRemoteSettings.xml` file, look for the following lines:

   ```
   <ProcessTypes>
   ```

2. For each object, set the `executeRemotely` variable to `true` for all process types you wish to be executed by the RPS. All other process types should be set to `false`.

Deploying the RPS Queue and Processor

The Remote Phoenix Server Queue (RPS Queue) is a Java servlet that runs in a Java servlet container. Tomcat must be installed on a computer that the JPS machine can ping. (Refer to the "Remote Processing Server" for more information.) The following discussion assumes that Tomcat is being used.

To deploy the RPS Queue

1. Install Apache Tomcat after the latest JDK is installed.

2. Shutdown Tomcat.

3. Copy `PhoenixServer.war` to `%TOMCAT_HOME%\webapps`.

4. Restart Tomcat.

The RPS Processor is installed on the same machine as the third party software (for NLME execution, no additional software is required). Currently, there are remote processors for NLME, NONMEM, PsN, R, and SAS. The RPS uses a plug-in architecture and it is possible to add processors not shipped with Phoenix.

Any number of RPS Processors can be configured and each one can be limited to processing specific job types.

To deploy the RPS Processor in Console mode

1. Copy `PhoenixJobProcessor.zip` to the machine that will perform the execution. The executing machine must be able to ping the RPS Queue machine.

2. Create a new folder where RPS Queue will be installed.

3. Unzip `PhoenixJobProcessor.zip` into the new installation folder.

4. Modify the `serverConfig.xml` file according to your needs. It must reference the RQS and have the correct executable path value specified. For example, `C:\Program Files\R\R-3.6.0\bin\R.exe` for R version 3.6.0. If the executing machine has multiple processors, the maxProcesses attribute can be increased to an appropriate number, and the RPS will process up to that many jobs simultaneously.

5. Valid values must be present in the `serverConfig.xml` for communicating with the Web server (DNS name or IP address), root (default is PhoenixServer), Web server port, SSL configuration, timeout, and pollinterval.

6. After saving changes to `serverConfig.xml` start the RPS Processor by opening a command window, navigating to the installation folder, and typing:
   `java -jar PhoenixJobProcessor.jar`

To deploy the RPS Processor as an Automatic Windows Service

1. Copy `PhoenixJobProcessor.zip` to the machine that will perform the execution. The executing machine must be able to ping the RPS Queue machine.

2. Create a new folder where RPS Queue will be installed.

3. Unzip `PhoenixJobProcessor.zip` into the new installation folder at the root of this folder.

4. Modify the serverConfig.xml file according to your needs. It must reference the RQS and have the correct executable path value specified. For example, `C:\Program Files\R\R-3.6.0\bin\R.exe` for R version 3.6.0. If the executing machine has multiple processors, the maxProcesses attribute can be increased to an appropriate number, and the RPS will process up to that many jobs simultaneously.

5. Valid values must be present in the serverConfig.xml for communicating with the Web server (DNS name or IP address), root (default is PhoenixServer), Web server port, SSL configuration, timeout and pollinterval.

6. Edit the file `install_service.bat` to reflect the location of the folder created in Step 3 for the entries `{set job_processor_home=}` and `{job_processor.exe //IS//PhoenixJobProcessor --Install=}` editing only the path statement after each of these parameters.

7. Save and close the batch file.

8. Execute the batch file as an administrative user on the target Windows system.

9. Optionally, users can edit the Windows Service to start automatically.

> ***Note:*** If you do not do so after each reboot, you must restart this service to enable the processing of jobs sent to the RPS Queue.

10. To edit the Windows Service startup option do the following:
    - Click **Start** or **Start Orb** and select the **Run** option.
    - Type `Services.msc` and press **Enter**. This opens the Services Console Window.
    - Find and double-click the Service named **Phoenix Job Processor** unless you edited the installation batch file to change this name.
    - Change the **Startup** type for this service to **Automatic** and click **Start**.

Troubleshooting

The Services can be invoked, if necessary, in **Console** mode via the following procedure. It is only necessary to invoke the Services in Console mode for troubleshooting.

1. Stop the service you want to run via command line in the Services console.

2. Open a command prompt window and navigate to the location of the Phoenix installation (e.g., for 64-bit machines: `C:\Program Files (x86)\Certara\Phoenix\application`).

3. Type the following text to start the Phoenix Job Processing Server:
   `jps.exe /console`

After pressing the **Enter** key, the Server will load and verify that it has access to a license to operate. If no named user license server is defined in the `jps.exe.config` file, it will not complete startup. The screen capture below illustrates a successful loading of the Job Processing Server.

Figure 3-1. Command prompt window showing successful loading of the JPS

To start the Phoenix Job Queue Server, the same steps are applied.

1. Open another command prompt window and navigate to the location of the Phoenix installation (e.g., for 64-bit machines: `C:\Program Files (x86)\Certara\Phoenix\application`).

2. Type the following text to start the Phoenix Job Queue Server:
   `jqs.exe /console`

The end result is that you have two command prompt windows open that can be minimized but not closed.

If a JMS component cannot pass licensing, it will not run as a Service. When run in Console mode, an entry in the command line interface will clearly show the user that a license could not be acquired.

Figure 3-2. Console window showing license error

JMS Admin Tool

Note: The JMS Admin Tool should only be installed on the machines of users who are going to act as JMS administrators. It is not included in a typical Phoenix installation.

This section includes the following topics:

Installing the JMS Admin Tool
Starting the Admin Tool
Changing job priority
Canceling a job

Installing the JMS Admin Tool
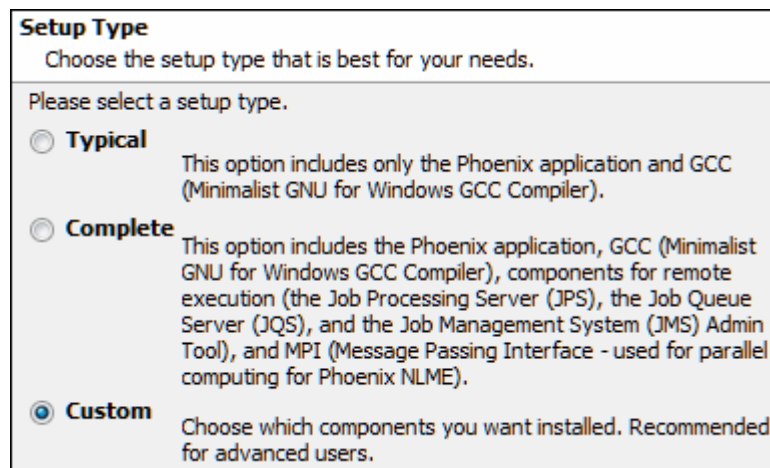
1. Select the **Custom** option during Phoenix installation.

Setup Type
Choose the setup type that is best for your needs.

Please select a setup type.

○ **Typical**
    This option includes only the Phoenix application and GCC (Minimalist GNU for Windows GCC Compiler).

○ **Complete**
    This option includes the Phoenix application, GCC (Minimalist GNU for Windows GCC Compiler), components for remote execution (the Job Processing Server (JPS), the Job Queue Server (JQS), and the Job Management System (JMS) Admin Tool), and MPI (Message Passing Interface - used for parallel computing for Phoenix NLME).

◉ **Custom**
    Choose which components you want installed. Recommended for advanced users.

Figure 4-1. Setup Type page for a custom installation

2. Expand the **Phoenix Application** node.

3. Select **Will be installed on local hard drive** for the **JMS Admin Tool** feature.

Custom Setup
Choose the program features you would like to install.

Click on an icon in the list below to change how a feature is installed.

Phoenix
    GCC
    JPS
    JQS
    JMS Admin Tool
    MPIC

Feature Description
The JMS Admin Tool allows users to update the priority of and/or cancel JMS jobs.

Will be installed on local hard drive
Entire feature will be installed on local hard drive
Entire feature will be unavailable

Figure 4-2. Custom Setup page showing the install options popup

4. Continue with Phoenix installation.

Starting the Admin Tool

1. Open Windows Explorer.

2. Navigate to the Phoenix installation directory (e.g., for 64-bit machines: `C:\Program Files (x86)\Certara\Phoenix\application`).

3. Double-click **JobManagementAdmin.exe**.

Once the application is started it will display a list of jobs currently in the JMS queue.

Figure 4-3. JMS Admin Tool

Jobs are displayed in the order that they will be executed with the job that will be executed first at the top of the list.

Changing job priority

1. Click the **Set the priority for the selected job** icon in the toolbar.
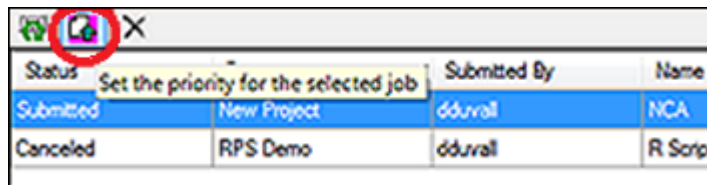
Figure 4-4. Click icon to change the priority

2. Use the *Priority* dialog to change the priority or promote the job within its priority.
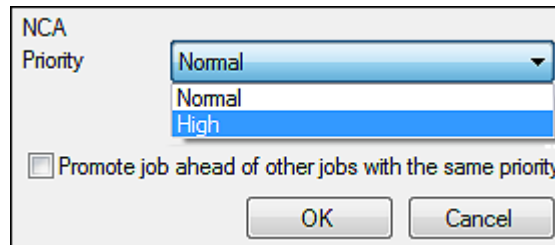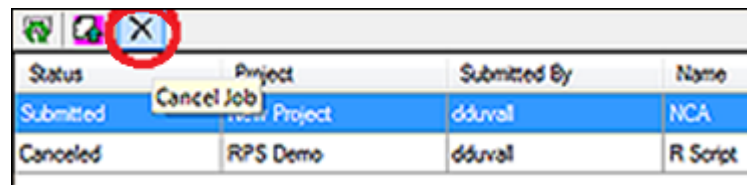
Figure 4-5. Priority dialog

3. Click the **Promote job...** checkbox to execute the job before other jobs with the same priority, even if they were submitted prior to the job. If more than one job within a priority have been promoted, then submission time will determine which will be processed first.

Canceling a job

1. Select the job to be canceled.

2. Click **Cancel Job** icon in the toolbar.

Figure 4-6. Click icon to cancel a job

Using JMS from within Phoenix

Setting JMS preferences

Access Job Management configuration within Phoenix via the *Preferences* dialog (**Edit > Preferences**).

In the *Preferences* dialog, expand **Remote Execution** and select **JMS**.

Check the **Remote Submit Enabled** box.

> If the checkbox is not checked, the **Remote Execute** icon in the Phoenix toolbar is present but disabled. When checked, the icon is enabled, allowing the user to send jobs to the JQS.
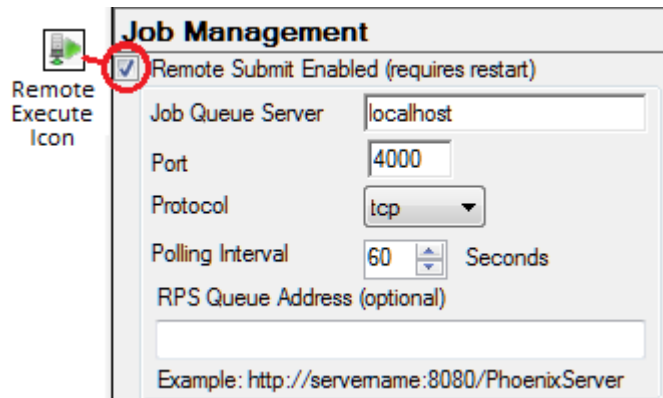
Figure 5-1. Job Management preferences

The **Job Queue Server** name (DNS Name or IP address), **Port**, and **Protocol** should be provided to you by you administrator.

In the **Polling Interval** field, enter how often you want your local client to sync up with the JQS.

In the **RPS Queue Address** field, optionally enter the fully qualified address of the RPS queue (e.g., `http://myserver:8080/PhoenixServer`).

How to submit a job

JMS is intended to facilitate the off-loading of Phoenix processing to a remote source allowing the user to continue working as large and time consuming workflows are processed. These workflows and executable objects are later merged in to the originating Phoenix project.

To perform a remote execution of an object, click the **Remote Execute** icon and select **Yes** in the *Remote Execution Confirmation* dialog.
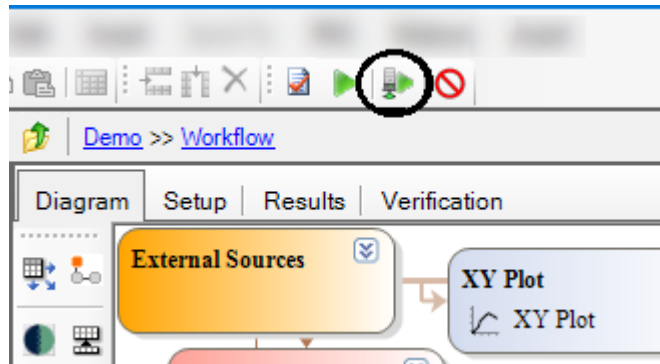
Figure 5-2. The Remote Execution icon is located in the Phoenix toolbar

The next dialog confirms successful submission to the remote Job Queue (JQS) of the job. Any executable that is not out-of-date or otherwise requiring executing cannot be submitted as a job.

If the remote machine cannot be reached, a dialog will display an error message and the locked objects will become unlocked.

Once the job is submitted, the user will not be permitted to make changes to any of the sources or dependents of the object(s) being executed. They are in read-only state so that changes that happen to the objects on the server can be merged into the project. The user can add new objects and multi-task within the project while the job is executing. The exception is if the root workflow of the project is submitted. Locked workflows do not allow insertion or deletion of executables.
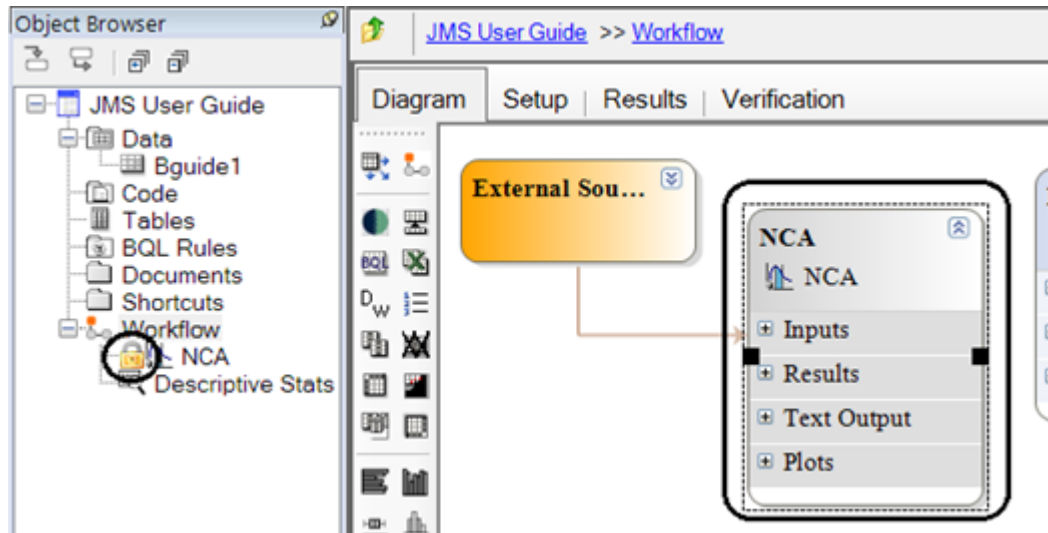
Figure 5-3. NCA has been submitted as a job, note the color in the diagram and the lock icon in the Object Browser

Viewing the history for any of the objects will reveal an entry showing that the object was executed remotely.

Figure 5-4. An entry in the history showing the remote execution of a Descriptive Statistics object

Note: Projects executed on the Job Management Service will report local times in the History for that object. If the Server and Workstation times are not in sync this could lead to events appearing to have occurred out of order. The system administrator should always attempt to keep all network participants time in sync.

A message in the Status Bar will notify the user once the remote execution completes. (Information also appears in the Status Bar when there is communication from the client to the JQS (e.g., sending job, receiving job results)).

Jobs Viewer

The *Jobs Viewer* is used to monitor jobs as they are being processed, and to merge results back into projects. The user can carry on with other work and at a later time check on the Jobs Viewer accessed via the **Window > View Jobs** (CTL+J). The *Jobs Viewer* displays several properties of the jobs including:

Status: Displays where the job is in the processing flow. The typical ordering of the states is.

> **Submitted**: Job is on the queue waiting to be processed.
> **Processing**: Job is being processed by a Job Processing (JPS) node. In this state, there will be additional information letting you know what machine it is being processed on and how long it has been running.
> **Results in Queue**: Job has finished processing and the results can be merged back into the original project.
> **Finished**: Job was completed and results merged back into the project.
> **Canceled**: Job was canceled by the user.

Project: The name of the project from which the job was submitted.

Submitted By: The username of the person who submitted the job.

Name: The name of the object submitted.

Source Project: The full path to the project from which the job was submitted.

Submit Time: The day, month, year, and time the job was submitted.

Processing Time: The length of computational time it took to complete the job.

Figure 5-5. Jobs Viewer listing all jobs submitted to the JMS (no filtering)

The checkboxes at the bottom of the viewer enable quick filtering of the listed jobs (**Filter out Finished/Canceled jobs** or **Filter out jobs from other users**). By default, only jobs submitted by the current user that are actively executing are listed.

Job Viewer toolbar icons include:

  ✕ Cancel the execution. (A confirmation dialog will display.)

  🔄 Refresh of the viewer list.

  ⬆ Merge a selected job.

The RPS queue can also be accessed via the *Jobs Viewer* through the **Action > View RPS Queue** menu item.

Merge results

Merging results requires that the user have the original project file, the project (or a copy of the project) is not currently opened in Phoenix, and nothing is currently executing.

When the submitted Workflow is at a Status of **Results in Queue**, click **Merge** (or select the **Action > Merge** menu item.)

In the *Job Results Merge* dialog, use the **Keep project open after merge** checkbox to indicate if the project remain open or be closed after merging. (The default is to keep it open).
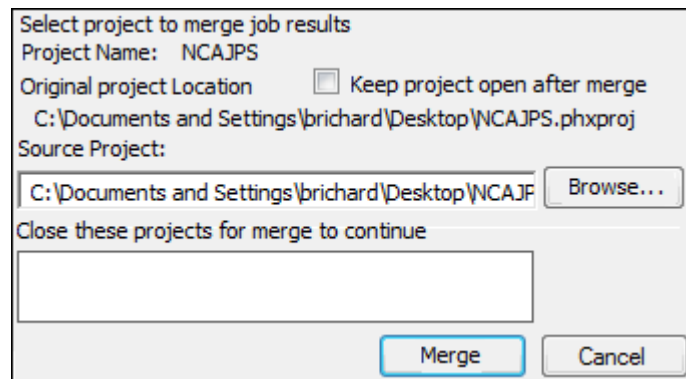
Figure 5-6. Job Results Merge dialog

Click **Browse** to select an alternate source project with which to reconcile the results.

If the source project, a project with the same name, or a project that has ID conflicts is already open in Phoenix, it is listed in the **Close these projects for merge to continue** area.

Click **Merge** to merge the results.

When the data merge is complete, the *Job Results Merge* dialog will close and the status of the merged entry will change to **Finished**.

Note: The results of a job can only be merged once.

Manage jobs within a project

In the event of canceling a job, the project the job was sent from remains locked. In this scenario, you will need to remove the job from the project to unlock the locked objects. This is done from the project root's property panel.
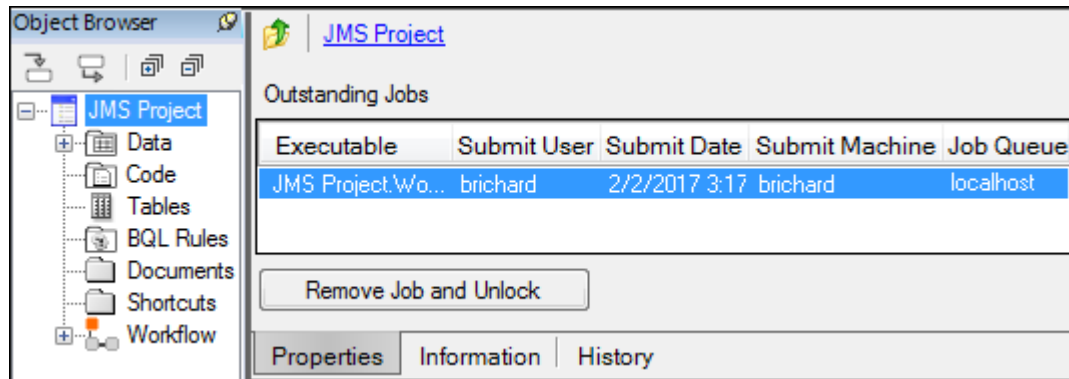

Figure 5-7. Properties panel of a project

Select the job to remove from the project and click **Remove Job and Unlock**.

This permanently removes the job information from the project and unlocks the objects. Only do this if you no longer wish to merge the jobs results back into the project. Once the project is saved this cannot be undone the deleted job will no longer merge into the project.

Note: You can only delete jobs that you have submitted from the JMS queue. Only the Admin has the ability to delete any job from the queue.