

# Phoenix<sup>®</sup> Remote Processing Server Administrator's Guide

Applies to:  
Phoenix 8.3

## 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 Dev-Components 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.

**Trademarks**

AutoPilot Toolkit, Integral, IVIVC Toolkit, JMS, Job Management System, NLME, Phoenix, PK Submit, Trial Simulator, Validation Suite, WinNonlin are trademarks or registered trademarks of Certara USA, Inc. NONMEM is a registered trademark of ICON Development Solutions. S-PLUS is a registered trademark of Insightful Corporation. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. Sentinel RMS is a trademark of Gemalto NV. Microsoft, MS, .NET, SQL Server Compact Edition, the Internet Explorer logo, the Office logo, Microsoft Word, Microsoft Excel, Microsoft PowerPoint®, Windows, the Windows logo, the Windows Start logo, and the XL design (the Microsoft Excel logo) are trademarks or registered trademarks of Microsoft Corporation. Pentium 4 and Core 2 are trademarks or registered trademarks of Intel Corporation. Adobe, Acrobat, Acrobat Reader, and the Adobe PDF logo are registered trademarks of Adobe Systems Incorporated. All other brand or product names mentioned in this documentation are trademarks or registered trademarks of their respective companies or organizations.

Certara, L.P.  
100 Overlook Center, Suite 101, Princeton, NJ, 08540 USA  
Telephone: +1.609.716.7900  
[www.certara.com](http://www.certara.com)

---

# Contents

- Remote Processing Server .....1**
- RPS Processor set up .....1
  - Windows Service Notes .....3
  - Linux Daemon Notes .....4
- RPS Queue set up .....5
- RPS Admin user management .....6
- RPS job viewer .....8



# Remote Processing Server

The RPS (Remote Processing Server) allows users to submit particular steps in their workflows to external programs (e.g., NONMEM, SAS, etc.) for remote execution. The RPS Queue and RPS Processor work together to efficiently submit the remote jobs to the various programs and return the results.

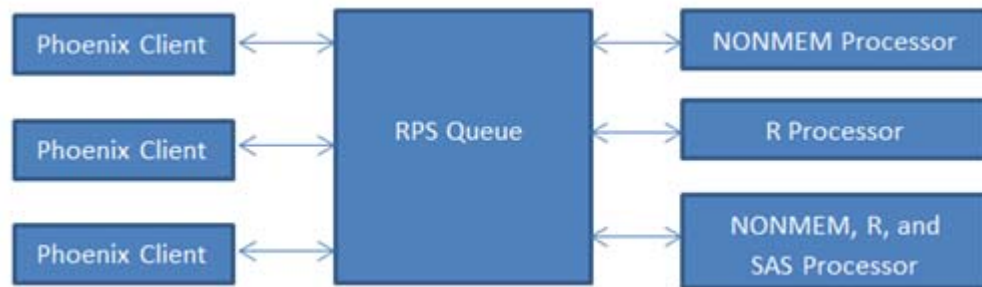


Figure 2-1. Diagram of how remote processing works

This section includes the following topics:

- [RPS Processor set up](#)
- [RPS Queue set up](#)
- [RPS Admin user management](#)
- [RPS job viewer](#)

## RPS Processor set up

Each RPS processing server should contain only one RPS processor installation, but each processor installation can be configured to process multiple jobs concurrently. RPS processors can be installed on any number of servers, and each processor can be configured to process different job types. For instance, there could be one server that only processes R jobs, one server that only processes NONMEM jobs, and yet another server that processes R, NONMEM and SAS jobs. The queue will only send a processor jobs of types that it is configured to process.

### *To install a processor*

1. Copy `PhoenixJobProcessor.zip` to the machine that will perform the execution. The executing machine must be able to ping the machine where the RPS Queue is installed.
2. Create a new folder where the RPS processor will be installed.
3. Unzip `PhoenixJobProcessor.zip` into the new installation folder.
4. Modify the `serverConfig.xml` file as needed (discussed below).
5. After saving changes to `serverConfig.xml` start the RPS by opening a command window, navigating to the installation folder and typing the following:

```
java -jar PhoenixJobProcessor.jar
```

See [“Windows Service Notes”](#) and [“Linux Daemon Notes”](#) for some platform-specific information.

RPS processors are configured by editing the file `serverConfig.xml`. Configuration consists of specifying the queue address, defining which types of processes the server can execute, specifying the number of concurrent processes the server is allowed to process, and process-specific attributes such as desired output location.

`settings` element configuration attributes:

`logLevel`: Required attribute. Valid values include: ALL, FINEST, FINER, FINE, CONFIG, INFO, SEVERE, WARNING, and OFF.

`processors` element configuration attributes:

`maxProcesses`: Required attribute. Maximum number of processes the server can execute simultaneously.

`maxLogLines`: Required attribute. Maximum number of lines to return when log info is requested by the Queue.

`processor` element configuration attributes:

`name`: Required attribute. Valid names include: `r`, `sas`, `nonmem7`, `psn`, and `custom`.

`class`: Required attribute. Java class to use to process all jobs for that type. The class must implement `com.Certara.phx.process.IProcessor`. This value should only change if a custom processor is required.

`allowConcurrentExecution`: Required attribute. Indicate if more than one instance of this process type can execute simultaneously. Values are `true` or `false`.

`logFile`: Not a required attribute. File name to use for job specific log information requested by the Queue. If not specified, no job specific log info will be available in the Queue.

`outputFolder`: Not a required attribute. Root location for output from this job type. Each job creates a unique subdirectory based on the Job ID. If not specified, a temporary folder is created and deleted after execution and results are returned to the Queue.

`outputShareName`: Not a required attribute. Name that the output folder is shared as on the network. This is useful when there are shortcut results. If the share name is specified, Phoenix shortcuts use the share name when constructing UNC paths to the file. If not specified, then a UNC path is constructed in the form: `\\server\PathFromRoot` for shortcut results.

`server` element configuration attributes:

`name`: Required attribute. RPS Queue Server name.

`root`: Required attribute. Context root of the Queue Server. Typically, this value is `Phoenix-Server`.

`port`: Required attribute. Port to use when accessing the Queue Server.

`useSSL`: Required attribute. Valid values are `true` and `false`.

`timeout`: Required attribute. The http timeout value in seconds.

`pollinterval`: Required attribute. Defines, in seconds, how often requests are sent to the Queue.

The example configuration file that is shipped with the product contains definitions for all process types. The types of processes that are processed on a server can be limited by removing or commenting unused process types. For example, if the processor is only to process NONMEM and R process types, then the file would look something like this:

```
<settings logLevel="CONFIG">
  <processors maxProcesses="1" maxLogLines="50">
```



```

    <processor name="r" class="com.Certara.phx.process.
RProcessor" allowConcurrentExecution="false"
    logfile="PHOENIX.LOG" outputFolder=
    "c:\\Output\\RPS\\R" outputShareName="R">
    <parameter name="executablepath" value=
    "C:\\Program Files\\R\\R-2.13.1\\bin\\R.exe"/>
    <parameter name="args" value="--quiet"/>
  </processor>
  <processor name="nonmem7" class="com.Certara.phx.
process.NonMemProcessor" allowConcurrentExecution=
  "false"
    logfile="stdout.txt" outputFolder=
    "c:\\Output\\RPS\\NONMEM">
    <parameter name="executablepath" value=
    "M:\\NMVII\\run\\phx.nmfe7.bat"/>
  </processor>
</processors>
<server name="localhost" root="PhoenixServer" port="8080"
  useSSL="false" timeout="300" pollInterval="5" />
</settings>

```

## Windows Service Notes

The `install_service.bat` file can be used to install the service. A few modifications might be required to the script to accommodate variations in specific installation file paths. The batch file specifies the necessary Java options required to run the PhoenixJobProcessor, and specifies a location in which to find the `serverConfig.xml` file (JobProcessorHome environment variable).

If a 64-bit version of Java is being used, replace all instances of `job_processor.exe` with `job_processor64.exe`.

### Example

```

@echo off

rem -----
rem Start/Stop Script for the Phoenix Job Processor
rem
rem -----

set job_processor_home=C:\ PhoenixRemote

set cp=%job_processor_home%\lib\apache-mime4j-0.6.jar
set cp=%cp%;%job_processor_home%\lib\commons-codec-1.3.jar
set cp=%cp%;%job_processor_home%\lib\commons-daemon-
1.0.2.jar
set cp=%cp%;%job_processor_home%\lib\commons-logging-
1.1.1.jar
set cp=%cp%;%job_processor_home%\lib\httpclient-4.0.1.jar
set cp=%cp%;%job_processor_home%\lib\httpcore-4.0.1.jar
set cp=%cp%;%job_processor_home%\lib\httpmime-4.0.1.jar
set cp=%cp%;%job_processor_home%\lib\xercesImpl.jar

```

```
set cp=%cp%;%job_processor_home%\lib\xml-apis.jar
set cp=%cp%;%job_processor_home%\PhoenixJobProcessor.jar

REM echo %cp%
@echo on

job_processor.exe //IS//PhoenixJobProcessor --Install=
C:\PhoenixRemote\ job_processor.exe --Description=
"Phoenix Job Processor" --Jvm=auto --Classpath=%cp% --
StartMode=jvm --StartClass=com.Certara.phx.process.
ProcessRunnerService --StartParams=start --StopMode=
jvm --StopClass=com.Certara.phx.process.
ProcessRunnerService --StopParams=stop --LogPath=
%job_processor_home%\logs --LogPrefix=service.log --
JvmOptions=-DJobProcessorHome="%job_processor_home%"
```

### Linux Daemon Notes

Use the script `phoenix_job_processor.sh` to run RPS as a Linux daemon. The script must be modified to update the values for `JAVA_HOME`, `PHOENIX_JOB_PROCESSOR_HOME`, `DAEMON_HOME`, `TMP_DIR`, and `JOB_PROCESSOR_USER` so that they are appropriate for the user's environment.

Once the script has been modified, start RPS like this:

```
sh ./phoenix_job_processor.sh start
```

Stop the process using the stop argument like this:

```
sh ./phoenix_job_processor.sh stop
```

### Example

```
#!/bin/sh
#####
#
#   Copyright 2010 Certara
#
#####
#
# Small shell script to start/stop Phoenix Job Processor
# using jsvc. Adapt the following lines to your
# configuration
JAVA_HOME=/usr/java/jre1.5.0_22
PHOENIX_JOB_PROCESSOR_HOME=/share/job_processor_daemon
DAEMON_HOME=/usr/java/jdk1.5.0_22/bin
JOB_PROCESSOR_USER=root
TMP_DIR=/var/tmp
CLASSPATH=\
lib/apache-mime4j-0.6.jar:\
lib/commons-codec-1.3.jar:\
lib/commons-daemon-1.0.2.jar:\
lib/commons-logging-1.1.1.jar:\
lib/httpclient-4.0.1.jar:\
```

```

lib/httpcore-4.0.1.jar:\
lib/httpmime-4.0.1.jar:\
lib/xercesImpl.jar:\
lib/xml-apis.jar:\
PhoenixJobProcessor.jar

case "$1" in
  start)
    #
    # Start Phoenix JobProcessor
    #
    cd $PHOENIX_JOB_PROCESSOR_HOME
    $DAEMON_HOME/jsvc \
    -user $JOB_PROCESSOR_USER \
    -home $JAVA_HOME \
    -Djava.io.tmpdir=$TMP_DIR \
    -outfile
    $PHOENIX_JOB_PROCESSOR_HOME/phoenix_job_processor.out \
    -errfile '&1' \
    -cp $CLASSPATH \
    com.Certara.phx.process.ProcessRunnerDaemon
    #
    # To get a verbose JVM
    #-verbose \
    # To get a debug of jsvc.
    #-debug \
    ;;

  stop)
    #
    # Stop Phoenix JobProcessor
    #
    PID=`cat /var/run/jsvc.pid`
    kill $PID
    ;;

  *)
    echo "Usage startup.sh start/stop"
    exit 1 ;;
esac

```

## RPS Queue set up

The RPS Queue is a Web Archive that is deployed to a servlet container. The following instructions describe how to deploy the RPS Queue to Tomcat.

1. Stop Tomcat.
2. Copy PhoenixServer.war to %TOMCAT\_HOME%\webapps.
3. Start Tomcat.

A few configuration options are available by editing the file %TOMCAT\_HOME%\webapps\PhoenixServer\WEB-INF\web.xml. Care should be taken when editing this file, and changes should only be made to the parameter values specified. The parameter names should never be changed.

Steps for configuring:

1. Stop Tomcat.
2. Make desired changes to %TOMCAT\_HOME%\webapps\PhoenixServer\WEB-INF\web.xml.
3. Save the file.
4. Start Tomcat.

Configurable parameter values include:

logLevel: Valid values in order of increasing detail are: CONFIG, INFO, WARNING, and SEVERE.

logPath: Relative path to the directory where logs should be written.

enableAdminSecurity: Valid values are true and false.

adminUserPath: Relative path to the XML file that contains the defined admin users.

### **Example**

```
...
<init-param>
  <param-name>logLevel</param-name>
  <param-value>INFO</param-value>
</init-param>
<init-param>
  <param-name>logPath</param-name>
  <param-value>../../logs</param-value>
</init-param>
...
<init-param>
  <param-name>enableAdminSecurity</param-name>
  <param-value>>true</param-value>
</init-param>
<init-param>
  <param-name>adminUserPath</param-name>
  <param-value>AdminUsers.xml</param-value>
</init-param>
...
```

## **RPS Admin user management**

Admin users are managed by editing the file AdminUsers.xml. The location of the file is configurable, but the default file location is %TOMCAT\_HOME%\webapps\PhoenixServer\AdminUsers.xml.

By default, an example user “admin” is in the shipped file.

```
<adminUsers>
  user name="admin" password=" " />
</adminUsers>
```

Notice that the password attribute is empty. The password attribute should always be left empty when adding a new user. The user will be prompted to create a password the first time they login. The password will be encoded when it is saved.

For instance, to define two administrators with user names johndoe and janedoe, the file would look like this:

```
<adminUsers>
  <user name="johndoe" password=" " />
  <user name="janedoe" password=" " />
</adminUsers>
```

Once the user has been added to the file, navigate to the Job Viewer.

Example address: <http://myhost:8080/PhoenixServer>

1. Click the **Login as administrator** link.



Figure 2-2. Phoenix Job Viewer showing "Login as administrator" link

2. Enter user name and leave the password blank.

The screenshot shows the "Admin Login" page of the Phoenix Job Server. The title "Phoenix Job Server" is at the top, followed by "Admin Login" in a bold font. There are two input fields: "User:" with the text "johndoe" entered, and "Password:" which is empty. Below these fields is a "Submit" button.

Figure 2-3. Administrator login page

3. Click **Submit**.
4. Enter the password twice and click **Submit**.



Figure 2-4. Change password page

Now, if the `AdminUsers.xml` file is opened, the an encoded password is listed for the user johndoe.

```
<user name="johndoe" password=" am9obmRvZQ==" />
```

To reset a password, delete the password text so that the password attribute is empty, and the user will once again be prompted to enter a password the next time they login.

Add and delete users by editing the file as necessary.

## RPS job viewer

RPS job details can be viewed by navigating to the Job Viewer in a browser. Users can get the URL from their internal IT personnel.

Example address: `http://servername:8080/PhoenixServer`

The Viewer will list the IDs, type, job submitter, status, user machine and JPS machine, time of submitted and the elapsed time, job priority, etc. When a job is processing, it is also possible to view a tail of the log file, provided the log file has been specified on the processing machine. When a job begins processing, a **View Log** button is displayed in the last column of the Job Viewer.

Clicking **View Log** will take the user to the Log Viewer window. The Log Viewer only displays the last several lines of the log. The number of lines displayed is configurable on the processing machine.

Administrators can login by clicking the link **Login as administrator**, entering their user name and password, and clicking **OK**. Once a user is logged in as administrator, they are able to kill jobs and update priorities for jobs that are waiting to process.

Jobs are processed first by priority and then by submit time. So, if two jobs are changed to **High** priority, the one that was submitted first will be the first job processed. All jobs are assigned the default priority of **Normal** when they enter the queue.

The **Kill** option, which is displayed in the last column of the Viewer for administrators, should only be used as a last resort. Users should first attempt cancel jobs using the Phoenix Job Viewer. Theoretically, it should never be necessary to use the **Kill** option. However, if for some reason a job becomes stuck in the queue or cannot be canceled from the client, the **Kill** option can be used to cancel and clear the job from the queue.