24x7 Scheduler™

Multi-platform Edition 7.5

Installing and Running 24x7 Web-Based Management Console

Table of Contents

About This Guide

This reference describes how to install, configure and run 24x7 Web-based Management Console with Apache Tomcat Servlet/JSP Container on all supported operating systems. This reference contains information for experienced users of the 24x7 Scheduler and assumes that you understand basic concepts of your operation system.

Conventions Used in This Document

This section describes the style conventions used in this document.

Italic
An *italic* font is used for filenames, URLs, emphasized text, and the first usage of technical terms.

`Monospace`
A `monospaced` font is used for code fragments and data elements.

Bold
A **bold** font is used for important messages, names of options, names of controls and menu items, and keys.

Graphical marks

 - This mark is used to indicate product specific options and issues and to mark useful tips.

 - This mark is used to indicate important notes.

Abbreviations and Terms

This guide uses common abbreviations for many widely used technical terms including FTP, HTTP, RAS, SQL, DBMS, SSH and other.

Trademarks

24x7 Automation Suite, 24x7 Scheduler, 24x7 Event Server, DB Audit, DB Audit Expert, SoftTree SQL Assistant are trademarks of SoftTree Technologies, Inc.

Apache Tomcat is a trademark of the Apache Software Foundation.

Windows 10, Windows XP are registered trademarks of Microsoft Corporation. UNIX is registered trademark of the X/Open Consortium. Sun, SunOS, Solaris, SPARC, Java are trademarks or registered trademarks of Sun Microsystems, Inc. Ultrix, Digital UNIX and DEC are trademarks of Digital Equipment Corporation. HP-UX is a trademark of Hewlett-Packard Co. IRIX is a trademark of Silicon Graphics, Inc. AIX is a trademark of International Business Machines, Inc. AT&T is a trademark of American Telephone and Telegraph, Inc.

All other trademarks appearing in this document are trademarks of their respective owners. All rights reserved.

Minimum Requirements

Software

1. 24x7 Scheduler Multi-platform Edition version 7.5 or later.

2. Java Development Kit (JDK) **JAVA_HOME** environment variable must be set to the pathname of the directory into which you installed the JDK release 1.8 or later.

3. Apache Tomcat Servlet/JSP Container (web server) version 9.0 or better or other compatible web server .

4. Optional Grafana server version 11.1 or later is required for using the **24x7 Performance Dashboard** feature.

5. **24x7 Web-based Management Console**

   Important Notes: 24x7 Web-based Management Console is optimized for use with 24x7 Scheduler Multi-platform Edition v7.5. While it can be used with other versions, not all console features might be available when working with older scheduler versions.

Hardware

1. Windows, Mac OS X, Linux or UNIX system capable of running Apache Tomcat web-server. See http://tomcat.apache.org/ for specific requirement

2. 4.8 MB free disk space for the 24x7 Web-based Console and 6 MB free disk space for compiled JSP files and temporary files.

3. 450 MB free disk space for the 24x7 Performance Dashboard and Grafana server powering the dashboard.

4. 8 GB RAM or better. Actual amount may vary for different operation systems and web server versions.

Web server and Performance Dashboard Installations

The installations are bundled with the 24x7 Scheduler Multi-platform Edition setup.

The following instructions are provided for scenarios in which a manual installation or a software upgrade is required.

Apache Tomcat Installation

Apache Tomcat Servlet/JSP Container version 9.0 or better can be used to run 24x7 Web-based Management Console. If you don't have Apache Tomcat installed, use the following 3 simple steps to download and install this software package:

1. Download and install the Tomcat Binary Distribution (either .ZIP or TGZ file) from http://tomcat.apache.org/

   Make sure to select version 9.0 or later.

2. Save the downloaded file into a temporary directory. The downloaded file name will have a name like jakarta-tomcat-[version].zip where [version] will be replaced with the Tomcat version number, for example, *jakarta-tomcat-9.0.28.zip*

   Important Notes:
   - In order to run the Tomcat web-server and the 24x7 Web-based Console you must have JDK 1.8 installed on the computer where the Tomcat web server will run. If you don't have it, download the required JDK from http://java.com/. Install the JDK according to the instructions included with the release.
   - Make sure **JAVA_HOME** environment variable is set to the pathname of the directory into which you installed the JDK release. See your operation system manual or search the web for instructions on setting environment variables.

3. After Tomcat Binary Distribution is downloaded, unzip it to the directory where you want the web-server to reside. The web server is now ready to be started

   Tips: Further information about configuring and running Tomcat can be found in the documentation included here with the package, as well as on the Tomcat web site http://jakarta.apache.org/tomcat/

Configuring Web Server

The following changes in the default Tomcat configuration are recommended for running 24x7 web console

1. Change **connectionTimeout** parameter: Locate **[Tomcat directory]/conf/server.xml** file. Open this file in text editor and search **connectionTimeout.** Change this value to 2000000000. Here is an example fragment of server.xml with this change.

   ```
   <Connector port="8088" protocol="HTTP/1.1"
            connectionTimeout="2000000000"
            redirectPort="8443" />
   ```

2. Set the start and maximum heap size for Tomcat server.
   - **set CATALINA_OPTS=-Xms256m -Xmx512m**   (Windows)
   - **export CATALINA_OPTS="-Xms256m -Xmx512m"** (UNIX, Linux, Mac OS X)

   It is recommended that you add this command to the web server's **startup file** or set it as a global environment variable to make this configuration change permanent. See the next topic for details on where to locate **startup.bat/sh** file.

Starting Web Server

Tomcat web server can be started by executing the following commands:

- **[Tomcat directory]\bin\startup.bat**   (Windows)

- **[Tomcat directory]/bin/startup.sh**   (UNIX, Linux, Mac OS X)

On Windows platforms, Tomcat can be installed and a run as a Windows service. In the latest case you do not need to use startup.bat. The service is started automatically for you.
After startup, the default Tomcat instance will be available by visiting http://localhost:8080/

 Important Notes: The default port used by Tomcat is port 8080 which it will be trying to bind on startup. In case if you have another web server or any other process using that port the Tomcat server will fail to start properly. To avoid that conflict configure Tomcat to some other port number. Edit **[Tomcat directory]/conf/server.xml** file and replace 8080 with some other port number which is not in use, for example, 8081. Restart Tomcat. If changes are made correctly the Tomcat instance will be available by visiting http://localhost:8081/ or whatever you have entered as the new port number.

Stopping Web Server

Tomcat web server can be stopped by executing the following commands:

- **[Tomcat directory]\bin\shutdown.bat**   (Windows)

- **[Tomcat directory]/bin/shutdown.sh**   (UNIX, Linux, Mac OS X)

24x7 Web-based Console Installation

Local Installation

Use the following simple steps to install 24x7 Web-based Console:

1. Copy 24x7 Web-based Console web archive file **24x7_wc_<version>.war** file to the computer running Tomcat. Here "version" represents numeric version and build number for the web console. Rename the downloader file to **scheduler.war**

2. Deploy 24x7 Web-based Console to the Tomcat web server. Several deployment methods are available. The following paragraphs describe 3 simple methods.

   Please note that third method seems to be easier it requires Tomcat restart which might be unacceptable if you use Tomcat with other applications.

- **Method 1 (recommended) – using the Tomcat's web application auto-deployment features:**

   Copy **scheduler.war** file to **webapps** directory. Wait for 10 seconds while Tomcat deploys the console application. That's it.

- **Method 2 (can be used in case the auto-deployment features are disabled – using the Tomcat's web application deployment tool or "Manager" application:**

The application deployment tool is available starting with Tomcat version 5, the "Manager" tool is available in most versions. Because of that we will review the latest method.

(a) Deploy the console application using the manager tool by entering the following URL into your web browser
http://localhost:8080/manager/deploy?path=/scheduler&war=jar:file:/**path_to_war**/scheduler.war!/

Here "path_to_ war" substring needs to be replaced with the absolute path to the directory where you have copied **scheduler.war** file.

if the deployment process completes successfully, you will see a web page with the following contents:

OK - Deployed application at context path /scheduler

Important Notes: The Tomcat Manager web-application may prompt you to enter user id and password. You must enter user and password that you have entered in the Tomcat **tomcat-users.xml** configuration file. If you haven't setup this file yet, edit **[Tomcat directory]/conf/tomcat-users.xml** with any text editor. This file contains an XML <user> for each individual user, which might look something like this:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="standard"/>
  <role rolename="manager"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user name="john" password="secret" roles="standard,manager"/>
</tomcat-users>
```

To enable access to the Manager web-application, you must either create a new username/password combination and associate the role name **manager** with it, or add the **manager** role to some existing username/password combination as in example above.

- **Method 3 manual installation – using the file system (can be used with all Tomcat versions):**

  (a) Rename **scheduler.war** to **scheduler.zip** and then unzip it to the **[Tomcat directory]/webapps/scheduler** directory. Be sure to unzip using *Restore Folder Names* option.

  (b) Restart Tomcat. Tomcat will automatically find the new web application, register it and start.

  To restart Tomcat you can use stop **shutdown** and **startup** commands described in the previous topics.

3. You can test the console is running by entering the following URL into your web browser
   http://localhost:8080/scheduler

Remote Installation

You will need to perform same basic steps that have been described in the Local Installation topic but in all URLs replace word "localserver" with the name of your web server computer, for example,

http://yourwebserver:8080/manager/deploy?path=/scheduler&war=jar:file:/**path_to_war**/jwi24x7.war!/

Upgrading Existing Installation

Copy new version of 24x7 Web-based Console web archive file **24x7_wc_<version>.war** file to the computer running Tomcat. Here "version" represents numeric version and build number for the web console. <mark>Rename the downloader file to **scheduler.war**</mark>

- **Method 1 (recommended) –   – using the Tomcat's web application auto-deployment features:**

  (a) Remove **scheduler.war** file from **webapps** directory. Wait for 10 seconds while Tomcat uninstalls the old version of the console application.
  (b) Copy **scheduler.war** file to **webapps** directory. Wait for 10 seconds while Tomcat deploys the ne version of the console application. That's it.

- **Method 2 – using the Tomcat web application deployment tool or "Manager" application:**

  (a) Undeploy the old version using the following command:
  http://localhost:8080/manager/undeploy?path=/scheduler

  or if doing it remotely
  http://yourwebserver:8080/manager/undeploy?path=/scheduler

  (b) Deploy the new version as described in the "Installation" topic.

- **Method 3 – manual installation using the file system:**

  Stop the console web-application using the following command:
  http://localhost:8080/manager/stop?path=/scheduler

  or if doing it remotely
  http://yourwebserver:8080/manager/stop?path=/scheduler

  (a) Copy or FTP **scheduler.war** to **[Tomcat directory]/webapps** directory overwriting the existing file.

  (b) Clear Tomcat cache for the application, deleting **[Tomcat directory]/work/Catalina/localhost/scheduler** directory .

  (c) Restart the application as described in the following topic.

Starting 24x7 Web-based Console

Normally after the deployment Tomcat automatically starts 24x7 Web-based Console. If for whatever reason it was stopped you can restart it using the following command:
http://localhost:8080/manager/start?path=/scheduler

You can test the console is running by using the following URL:

http://localhost:8080/scheduler

or if you are doing it remotely
http://yourwebserver:8080/scheduler

Troubleshooting

Why do I get the error "Bind exception: Address already in use" when I try to startup Tomcat?

The most probable reason is that you have an application conflict, in other words, an instance of Tomcat or some other application is already active in your system and using the default Tomcat port number 8080.

To avoid that conflict configure Tomcat to some other port number. Edit **[Tomcat directory]/conf/server.xml** file and replace 8080 with some other port number which is not in use, for example, 8081. Restart Tomcat. If changes are made correctly the Tomcat instance will be available by visiting http://localhost:8081/ or whatever you have entered as the new port number.

Why does the memory usage increase when I redeploy 24x7 Web-based Console?

Because the Tomcat's Classloader (and the Class objects it loaded) cannot be recycled. They are stored in the permanent heap generation by the JVM, and when you redeploy 24x7 web-based console a new class loader is created, which loads another copy of all these classes. This can eventually cause **OufOfMemoryErrors** errors. Restart Tomcat server to avoid dealing with this issue.

Why does Tomcat hang after starting on a Linux computer?

The most likely cause of such hang is the incompatibility of JDK with the new implementation of POSIX library on Red Hat Linux Systems. Multithreading applications like application servers are likely to face threading issues when using the new NPTL implementation on Linux.

To fix this issue, an environment variable LD_ASSUME_KERNEL needs to be exported before starting Tomcat. Add the following line to Tomcat's startup.sh script.

```
export LD_ASSUME_KERNEL=2.4.1
```

Why do I get "404 Error – page not found" while running 24x7 Web-based Console?

The most likely cause of such error is that some console files have been deleted from the deployment directory or haven't been installed. Try redeploying the application as described in the "Upgrading existing installation" topic.

Why do I get "Connection denied" error when logging into the 24x7 Web-based Console?

There are 4 likely causes:
- The Remote Control option has not been enabled in the 24x7 Scheduler settings. If you are running 24x7 Scheduler Windows Edition ensure the Master mode (the "distributed network mode" option) has been enabled.

- The 24x7 Scheduler doesn't run under privileged user account as a result is unable to authenticate the provided user credentials with the host operation system
- The security system has been enabled in the 24x7 Scheduler but the provided user is not added to a privileged group or unknown to the scheduler
- If you have a firewall between the web server and the scheduler computer, the firewall blocks TCP port used by the console and the scheduler for communications. (24x7 Scheduler Multi-platform Edition by default uses port 1097 while 24x7 Scheduler for windows uses port 1096). Be sure to open that port in the firewall for the network traffic flowing between the web server and the scheduling server computers.