

VSI Tomcat Version 9.0-84B for OpenVMS IA-64 and x86-64 (based on Apache Tomcat)

Release Notes

Publication Date: January 2025

Operating Systems: VSI OpenVMS Integrity Version 8.4-2L1 or higher
VSI OpenVMS x86-64 Version 9.2-2 or higher

Software Version: VSI Tomcat Version 9.0-84B for OpenVMS IA-64 and x86-64

Kit Names: VSI-I64VMS-VSI_TOMCAT-V0900-84B-1.PCSI
VSI-X86VMS-VSI_TOMCAT-V0900-84B-1.PCSI

Table of Contents

1. Introduction	3
2. Significant Changes	3
3. What Else is New in This Release?	3
4. Software Prerequisites	5
5. Documentation	6
6. Before You Install	6
7. Installing VSI Tomcat	6
7.1. Process Quotas	8
7.2. Installing in an Alternative Location	8
8. Configuring VSI Tomcat	8
9. Common Problems	9
10. Notes on Using CGI With Tomcat on OpenVMS	9

1. Introduction

VSI Tomcat is based on the open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSockets. Tomcat provides a pure Java HTTP web server environment in which Java code can run. For more information, refer to the Apache Tomcat project website .

VSI Tomcat Version 9.0-84B for OpenVMS is a full binary release of Apache Tomcat 9.0.84 for the OpenVMS operating system.

2. Significant Changes

This release of VSI Tomcat for OpenVMS includes significant changes from previous releases of VSI Tomcat for OpenVMS that are intended to enhance performance and reliability as well as to simplify administration. These changes are as follows:

Not dependent on Apache HTTP

Previous releases of VSI Tomcat for OpenVMS installed by default into a directory hierarchy that assumed Tomcat would be installed and used in conjunction with the Apache HTTP web server. In this release, Tomcat is installed into its own directory hierarchy with the top-level directory being defined by the TOMCAT\$ROOT logical name. With the exception of start-up and shutdown files, all files reside under this root. The web server logs reside in the TOMCAT\$ROOT:[LOGS] directory.

Simplified OpenVMS administration interface

Previous releases of VSI Tomcat for OpenVMS included a command procedure that could be used to configure ownership of the VSI Tomcat directory tree and a username under which the web server would run. This command procedure is no longer provided, and it is up to the system administrator to manually change file ownership and to specify at start-up (via a logical name or via a start-up parameter) the username under which the web server will run. Command procedures are provided to start up and shutdown the web server; a number of facilities are provided to allow the specification of site-specific details, such as Java and DECC logical names and site-specific JVM command line arguments.

3. What Else is New in This Release?

VSI Tomcat Version 9.0-84B is available only for the IA-64 and x86-64 architectures and includes the following changes (for more details, see <http://tomcat.apache.org/tomcat-9.0-doc/index.html>):

Dependency changes

Tomcat 9.0 is designed to run on Java SE 8 and later.

API stability

The public interfaces for all classes in the javax namespace are fixed and will *not* be changed at all during the remaining lifetime of Tomcat 9.

The public interfaces for the org.apache.catalina.* class (excluding sub-packages) may be added to in order to resolve bugs and/or add new features. No existing interface method will be removed or changed, although it may be deprecated.

As development of Tomcat 9 continues, the above information will be updated. This list is *not* considered complete at this time.

Note

A large number of deprecated methods, fields, and configuration options were removed in the transition to version 9. If any of those removals trigger significant problems for the user community, then the deletion may be reverted in a later release.

The remaining classes are considered part of the Tomcat internals and may change without notice between point releases.

Bundled APIs

A standard installation of Tomcat 9.0-84B includes all of the following APIs available for use by web applications (by placing them in the LIB directory):

annotations-api.jar	catalina.jar
catalina-ant.jar	catalina-ha.jar
catalina-ssi.jar	catalina-storeconfig.jar
catalina-tribes.jar	commons-el.jar
ecj-4.20.jar	el-api.jar
jasper.jar	jasper-el.jar
jspic-api.jar	jsp-api.jar
servlet-api.jar	taglibs-standard-compat-1.2.5.jar
taglibs-standard-impl-1.2.5.jar	taglibs-standard-jstlel-1.2.5.jar
taglibs-standard-spec-1.2.5.jar	tomcat-api.jar
tomcat-coyote.jar	tomcat-dbcp.jar
tomcat-i18n-es.jar	tomcat-i18n-de.jar
tomcat-i18n-es.jar	tomcat-i18n-fr.jar
tomcat-i18n-ja.jar	tomcat-i18n-ko.jar
tomcat-i18n-pt-BR.jar	tomcat-i18n-ru.jar
tomcat-i18n-zh-CN.jar	tomcat-jdbc.jar
tomcat-jni.jar	tomcat-util.jar
tomcat-util-scan.jar	tomcat-websocket.jar
websocket-api.jar	

You can make additional APIs available to all of your web applications by putting the unpacked classes into a CLASSES directory (note that it is *not* created by default), or by placing them in JAR files in the LIB directory.

To override the XML parser implementation or interfaces, use the appropriate feature of your JVM. For Java 8 or earlier, use the endorsed standards override feature. The default configuration defines the JARs located in the ENDORSED directory as endorsed. For Java 9+, use the upgradeable modules feature.

Web application reloading and static fields in shared libraries

Some shared libraries (many are part of the JDK) keep references to objects instantiated by the web application. To avoid problems with class loading (ClassCastExceptions, messages indicating that the classloader is stopped, etc.), the shared libraries state must be reinitialized.

Avoid putting classes that would be referenced by a shared static field in the web application classloader, and put them in the shared classloader instead; JARs must be put in the LIB folder, and classes should be put in the CLASSES folder.

Security manager URLs

In order to grant security permissions to JARs located in the web application repository, make sure that the URLs in your policy file comply with the following format:

```
FILE:${CATALINA.BASE}/WEBAPPS/EXAMPLES/WEB-INF/LIB/DRIVER.JAR
```

Symlinking static resources

By default, Unix symlinks will *not* work when used in a web application to link resources located outside the web application root directory.

This behaviour is optional, and the "allowLinking" flag may be used to deactivate the check.

JSP Standard Tag Library

This release contains the previously missing JSP Standard Tag Library (JSTL). This is a legacy library that is required for WSIT to be able to build and execute an out-of-process WSI-based application.

Viewing the Tomcat Change Log

The full change log is available at <https://tomcat.apache.org> and is included in the documentation for the web application of Tomcat.

Cryptographic Software Notice

This distribution includes cryptographic software. The country in which you currently reside may have restrictions on the import, possession, use, and/or re-export of encryption software. *Before* using any encryption software, check your country's laws, regulations, and policies concerning the import, possession, or use, and re-export of encryption software. For more information, see <http://www.wassenaar.org/>.

The U.S. Government Department of Commerce, Bureau of Industry and Security (BIS), has classified this software as Export Commodity Control Number (ECCN) 5D002.C.1, which includes information security software using or performing cryptographic functions with asymmetric algorithms. The form and manner of this Apache Software Foundation distribution makes it eligible for export under the License Exception ENC Technology Software Unrestricted (TSU) exception (see the BIS Export Administration Regulations, Section 740.13) for both object code and source code.

Note

Tomcat includes the code designed to work with JSSE and the code designed to work with OpenSSL.

For further information, see the FAQ at <https://tomcat.apache.org/faq/>.

4. Software Prerequisites

Before you install VSI Tomcat Version 9.0-84B for OpenVMS IA-64 and x86-64, make sure the following prerequisites are met:

- OpenVMS Integrity Version 8.4-2L1 or higher
- OpenVMS x86-64 Version 9.2-2 or higher
- Java™ Development Kit (JDK) 8 for the OpenVMS Integrity servers (VSI Tomcat V9.0-84B will *not* run with Java 7.)
- VSI Tomcat *must* be installed on an ODS-5-enabled disk.

5. Documentation

For information about Tomcat, see the Apache Tomcat 9 documentation at <http://tomcat.apache.org/tomcat-9.0-doc/index.html>.

6. Before You Install

Before installing VSI Tomcat V9.0-84B, you *must* shutdown the web server and manually remove any existing version of VSI Tomcat that is installed on your system. Be sure to backup any site-specific files before performing this operation.

The command to use to remove the existing kit depends on the kit version:

- If you have VSI Tomcat V8.5-88 or earlier installed, remove it via the following command:

```
$ PRODUCT REMOVE CSWS_JAVA
```

- If you have VSI Tomcat V8.5-89 installed, remove it via the following command:

```
$ PRODUCT REMOVE VSI_TOMCAT
```

Important

After removing the old version of Tomcat, you must also check for any Tomcat startup, shutdown, or customization files in SYS\$STARTUP and delete them.

The Apache Tomcat Native Library is an optional component for use with Apache Tomcat that allows Tomcat to use OpenSSL as a replacement for JSSE to support TLS connections. VSI Tomcat Version 9.0-84B has this option enabled by default. If you prefer not to use it, comment out the following row in the config/server.xml:

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"  
  SSLEngine="on" />
```

The native shareable image libtcnative-1.exe (version 1.2.23) can be found in the TOMCAT\$ROOT/SBIN directory.

7. Installing VSI Tomcat

The kits are provided as OpenVMS PCSI kits (VSIH64VMS-VSI_TOMCAT-V0900-84B-1.PCSI or VSI-X86VMS-VSI_TOMCAT-V0900-84B-1.PCSI) that can be installed by a suitably privileged user (SYSPRV privileges are required to install the product; SYSPRV, IMPERSONATE, and BYPASS privileges are required to run it) using the following command:

```
$ PRODUCT INSTALL VSI_TOMCAT
```

The installation will then proceed as follows (output may differ slightly from that shown):

```
$ PRODUCT INSTALL VSI_TOMCAT

Performing product kit validation of signed kits ...
The following product has been selected:
    VSI X86VMS VSI_TOMCAT V9.0-84B          Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.

Configuring VSI X86VMS VSI_TOMCAT V9.0-84B: Tomcat V9.0-84B-1 for VSI OpenVMS x86

    VMS Software Inc. & The Apache Software Foundation.

    VMS Software Inc.

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
    VSI X86VMS VSI_TOMCAT V9.0-84B          DISK$X86SYS:[SYS0.SYSCOMMON.]

Portion done: 0%...90%...100%

The following product has been installed:
    VSI X86VMS VSI_TOMCAT V9.0-84B          Layered Product

VSI X86VMS VSI_TOMCAT V9.0-84B: Tomcat V9.0-84B-1 for VSI OpenVMS x86

Post-installation tasks are required.

    By default to run Tomcat 9.0-84B OPENJDK 8.0 is used. Make sure that it
    was installed.
    To start the Tomcat web server at system boot time, add the following
    lines to SYS$MANAGER:SYSTARTUP_VMS.COM:

    $ file := sys$common:[sysmgr]TOMCAT$DEFINE_LOGICALS.com
    $ if f$search("''file'") .nes. ""
    $ then
    $     @'file'
    $     @tomcat$root:[000000]tomcat$startup.com
    $ else
    $     write sys$output "Couldn't find TOMCAT$ROOT definition
    $ endif

    To shutdown Tomcat at system shutdown time, add the following lines to
    SYS$MANAGER:SYSHUTDOWN.COM:

    $ file := sys$common:[sysmgr]TOMCAT$DEFINE_LOGICALS.com
    $ if f$search("''file'") .nes. ""
    $ then
    $     @'file'
    $     @tomcat$root:[000000]tomcat$shutdown.com
    $ else
    $     write sys$output "Couldn't find TOMCAT$ROOT definition
    $ endif

Note that default installation uses the SYSTEM account to run the the
Web server process. It is recommended that you run the web server as
using a less privileged account. This may be done by supplying the
```

account name as a parameter to tomcat\$startup.com or by defining the logical name tomcat\$user as the desired account name. It is also recommended that you change the tomcat\$root:[000000...] directory tree ownership to this account.!

7.1. Process Quotas

VSI Tomcat can require considerable resources in order to operate efficiently, depending on workload requirements. The following quotas should be adequate for most purposes; however resource usage should be carefully monitored, and quotas adjusted as necessary.

Maxjobs:	0	Fillm:	32767	Byt1m:	3000000
Maxacctjobs:	0	Shrfillm:	0	Pbyt1m:	0
Maxdetach:	0	BIO1m:	1024	JTquota:	40000
Prclm:	100	DI01m:	1024	WSdef:	100000
Prio:	4	AST1m:	300	WSquo:	200000
Queprio:	4	TQE1m:	400	WSextent:	800000
CPU:	(none)	Enqlm:	32767	Pgflquo:	10000000

If the web server is expected to support large numbers of connections then it may also be necessary to increase the CHANNELCNT system parameter (this parameter can usually be safely set to its maximum value of 65535).

It is also recommended to maintain a pagefile size of at least 20000000 blocks.

7.2. Installing in an Alternative Location

By default, VSI Tomcat will be installed in SYSSYSDEVICE:[VMS\$COMMON]. If you wish to install the software in an alternative location, use the **/DESTINATION** qualifier with the **PRODUCT INSTALL** command to specify the desired location.

8. Configuring VSI Tomcat

After the installation is complete, you may wish to perform the following configuration steps:

- Use the **SET FILE/OWNER** command to change the ownership of the TOMCAT\$ROOT directory tree to a user less privileged than SYSTEM. Ensure that the specified user exists and has the process quotas required to run Tomcat.
- Configure any site-specific VSI Tomcat and JVM options by adding the site-specific logical names and JVM command line options to the files SYSSSTARTUP:TOMCAT\$DEFS_LOCAL.COM and SYSSSTARTUP:TOMCAT\$ARGS_LOCAL.DAT, respectively.
- Start the web server and ensure that it is up and running. Note that the IMPERSONATE privilege is required by default in order to start the web server.

```
$ @SYSSSTARTUP:TOMCAT$STARTUP.COM
$ PIPE SHOW SYSTEM | SEARCH SYS$INPUT APACHE$TOMCAT
0000806D APACHE$TOMCAT  CUR   6  6   32178   0 00:00:29.54   39658  42146 M
```

Note that you should give the APACHE\$TOMCAT process a few seconds to start before checking that it process exists. If the VSI Tomcat web server does not start, check the log files in TOMCAT\$ROOT:[LOGS] for additional information.

- By default, the web server will run under the SYSTEM account. To run the process as a different user, either define the TOMCAT\$USER logical name before running the start-up procedure or specify the desired username as a parameter to the start-up procedure. As noted previously, be aware that IMPERSONATE privilege is required by default in order to start the web server.

Note that the TOMCAT\$USER logical name can be defined at the process level; it is only required by the Tomcat start-up procedure, SYS\$STARTUP:TOMCAT\$STARTUP.COM.

- After you have successfully configured and started VSI Tomcat, access the included JSP and servlet examples via <http://hostname:8080> to verify that the web server is operating correctly (replace *hostname* with the name or the IP address of the OpenVMS server on which Tomcat has been installed).

9. Common Problems

This section contains notes about the common problems that may be encountered when using VSI Tomcat on OpenVMS.

Access to Tomcat server is slow when it is invoked for the first time

When you invoke the Tomcat server for the first time, there might be a delay in accessing <http://hostname:8080>. The reason for the delay is that Tomcat deploys all of the applications (mostly examples) in the webapps directory. This deployment is done only when the server is invoked for the first time.

Redeploying .WAR file fails

Previously, to redeploy .WAR files, you had to delete all existing .WAR files and the directory tree where the webapp was executed. This was due to Tomcat's inability to delete multiple files and directories. Now, you can overcome this problem by defining the following logical names in the site-specific configuration file SYS\$STARTUP:TOMCAT\$DEFS_LOCAL.COM:

```
$ DEFINE JAVA$DELETE_ALL_VERSIONS 1
$ DEFINE JAVA$CREATE_DIR_WITH_OWNER_DELETE 1
```

Initially deploying .WAR files

It has been observed that .WAR files sometimes do not expand upon deployment. The reasons for this and the exact circumstances in which this may happen are being investigated; however, the evidence suggests that the problem is related to file ownership and permissions. A simple solution to the problem is to issue the following command:

```
$ SET FILE/PROTECTION=(W:RE) filename.WAR
```

After a short delay, Tomcat will begin to expand the files.

10. Notes on Using CGI With Tomcat on OpenVMS

The CGI (Common Gateway Interface) defines a way for a web server to interact with external content-generating programs, which are often referred to as CGI programs or CGI scripts.

By default, CGI support is disabled in Tomcat. To enable CGI Support, perform the following tasks:

1. Edit the TOMCAT\$ROOT:[CONF]WEB.XML file. Locate the existing (but commented out) section that defines the CGI environment and uncomment the servlet configuration section. For example:

```
<!-- -->
<servlet>
```

```
<servlet-name>cgi</servlet-name>
<servlet-class>org.apache.catalina.servlets.CGIServlet</servlet-class>
  <init-param>
    <param-name>clientInputTimeout</param-name>
    <param-value>100</param-value>
  </init-param>
  <init-param>
    <param-name>debug</param-name>
    <param-value>6</param-value>
  </init-param>
  <init-param>
    <param-name>executable</param-name>
    <param-value></param-value>  <!--Blank this value,  if using DCL or
Executables for CGIs.-->
  </init-param>
  <init-param>
    <param-name>cgiPathPrefix</param-name>
    <param-value>WEB-INF/cgi</param-value>
  </init-param>
  <load-on-startup>5</load-on-startup>
</servlet>
```

Note that if the parameter-name `executable` and the parameter-value resources are missing, you must add these in manually as shown above.

```
<init-param>
  <param-name>executable</param-name>
  <param-value></param-value>
</init-param>
```

If you are going to be using DCL or native executables as CGI programs, you must leave the `param-value` blank as seen above, otherwise you may receive an error message, and the CGI scripts will fail to work.

2. Next you must uncomment the `cgi` servlet-mapping definitions located just a little further down in the file:

```
<!-- The mapping for the CGI Gateway servlet -->
<!-- -->
  <servlet-mapping>
    <servlet-name>cgi</servlet-name>
    <url-pattern>/cgi-bin/*</url-pattern>
  </servlet-mapping>
```

After making this change, save your changes and exit the editor back to DCL.

3. Locate the file `TOMCAT$ROOT:[CONF]CONTEXT.XML` and edit it. Look for the line that reads `<Context>` and modify it to include the following directives:

```
<Context reloadable="true" privileged="true" >
```

4. Once you have these files modified to allow CGI scripting to function within VSI Tomcat on OpenVMS, you must create the following directory path, replacing *directory-owner* with the UIC or username of the owner of the `TOMCAT$ROOT` directory tree:

```
$ SET DEFAULT TOMCAT$ROOT:[WEBAPPS]
$ CREATE/DIRECTORY/OWNER=[directory-owner] [.CGI.WEB-INF.CGI]
```

5. After making these changes and restarting the web server, you should be able to run CGI scripts from the CGI directory using URLs of the following form:

<http://hostname:8080/CGI/cgi-bin/test.cgi>