



CSWS_JAVA for OpenVMS (based on Apache Tomcat)

Installation Guide and Release Notes

December 2015

Note that CSWS_JAVA Version 7.0-29B supplied by VMS Software Inc. is equivalent to the Version 7.0-29 kit provided by HP, but does not include the Apache HTTP Server connector modules MOD_JK and MOD_JK2.

What is New

CSWS_JAVA Version 7.0-29B is available on OpenVMS Integrity server. Apache Tomcat 7.0.29B improves on previous releases in the following ways:

- Security fixes (CVEs) are added to this release. For the list of security fixes, see the Apache Tomcat website at <http://tomcat.apache.org>.
- Added support for searching roles in JNDI/LDAP using a different value other than the actual DN or username specified.
- Added stuckThreadNames property as a pair for the stuckThreadIds one; add thread IDs to the log messages.
- Added support for the default error page to be defined in web.xml by defining an error page with a nested location element.
- Annotation scanning is now always performed regardless of the version declared in the web.xml file, expect metadata complete is set to true.

Overview

CSWS_JAVA includes the following projects:

- Tomcat (Catalina)
- Ant

For more information about Tomcat and other projects, see the Apache Tomcat project web site (<http://tomcat.apache.org/>).

Tomcat

CSWS_JAVA includes the following Apache Tomcat technologies:

- JavaServer Pages 2.2
- Java Servlet 3.0

Tomcat is the reference implementation for the Java Servlet 3.0 and JavaServer Pages 2.2 technologies. CSWS_JAVA includes the final Apache Tomcat V7.0.29.

Tomcat is a servlet container with a JSP environment. A servlet container is a runtime shell that manages and invokes Java servlets on behalf of users. Servlet containers can be standalone, in-process, or out-of-process. CSWS_JAVA includes support for standalone servlet containers and out-of-process servlet containers. Support for in-process servlet containers (JSSI) will be included in a future version of Tomcat.

For more information about Tomcat 7, see <http://tomcat.apache.org/tomcat-7.0-doc/index.html>.

Ant

Ant is also included in CSWS_JAVA. Ant is a partial implementation of the Jakarta Ant subproject, and its use is limited to building the included sample web applications and simple user-written web applications for Tomcat.

Software Prerequisites

CSWS_JAVA for OpenVMS requires the following software:

- OpenVMS Integrity servers Version 8.4-1H1
- Software Development Kit (SDK) for the OpenVMS Operating System, for the Java™ Platform Version 1.6.0
- All SDK 1.6.0 patches required for OpenVMS 8.4-1H1
- It is required that you install CSWS_JAVA on an ODS-5 enabled disk

Documentation

For information about Tomcat, see the Jakarta Apache Project and Tomcat 7 documentation, which can be found at <http://tomcat.apache.org/tomcat-7.0-doc/index.html>.

Before Beginning the Installation

Before you install the CSWS_JAVA kit, perform the following steps:

1. Shut down Tomcat if it is currently installed and running.

```
$ @SYS$STARTUP:APACHE$JAKARTA_SHUTDOWN
```

2. Remove earlier version of CSWS_JAVA, if installed.

Before installing CSWS_JAVA Version 7.0-29B, you must manually remove any existing version of CSWS_JAVA if it is installed on your system.

Use PCSI to remove CSWS_JAVA by entering the following command, and enter YES to the "Delete the Jakarta Ant & Tomcat directory trees" question.

```
$ PRODUCT REMOVE CSWS_JAVA
```

```
Delete the Jakarta Ant & Tomcat directory trees ? [NO]: YES
```

Installing CSWS_JAVA

VSI requires that you install CSWS_JAVA on an ODS-5 enabled disk.

Verify that the destination device is an ODS-5 volume by entering a command similar to the following (assuming \$2\$DKB400 is the disk where you want to install CSWS_JAVA):

```
$ SHOW DEV $2$DKB400 /FULL
```

```
Disk $2$DKB400: (NODE11), device type COMPAQ BD018635C4, is online,
mounted, file-oriented device, shareable, served to cluster via MSCP
Server, error logging is enabled.
```

```
.
```

```
Volume Status: ODS-5, subject to mount verification, file high-water marking
write-back caching enabled.
```

Install the CSWS_JAVA kit by entering the following command, where \$2\$DKB400 is the name of the ODS-5 enabled disk where you want to install CSWS_JAVA. Be sure that you manually removed the earlier version of CSWS_JAVA before proceeding.

```
$ PRODUCT INSTALL CSWS_JAVA/DEST=$2$DKB400:[000000]
```

For a description of the features you can request with the `PRODUCT INSTALL` command when starting an installation such as running the IVP, purging files, and configuring the installation, see the **POLYCENTER Software Installation Utility User's Guide**.

As the installation procedure progresses, the system displays the following information:

```
$ PRODUCT INSTALL CSWS_JAVA/DEST=$2$DKB400:[000000]
```

```
The following product has been selected:
```

```
VSI I64VMS CSWS_JAVA V7.0-29B          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 I64VMS CSWS_JAVA V7.0-29B
```

```
VMS Software Inc. & The Apache Software Foundation.
```

```
* This product does not have any configuration options.
```

```
Execution phase starting ...
```

```
The following product will be installed to destination:
```

```
VSI I64VMS CSWS_JAVA V7.0-29B          DISK$I64SYS:[VMS$COMMON.]
```

```
Portion done: 0%...10%...20%...30%...40%...50%...90%...100%
```

```
The following product has been installed:
```

```
VSI I64VMS CSWS_JAVA V7.0-29B          Layered Product
```

```
VSI I64VMS CSWS_JAVA V7.0-29B
```

```
Post installation tasks required for CSWS_JAVA for OpenVMS
```

```
Configure OpenVMS aspects of CSWS_JAVA by:
```

```
$ @SYS$MANAGER:APACHE$JAKARTA
```

```
The default installation uses the SYSTEM account to run the CSWS_JAVA
(Jakarta/Tomcat) engine. If you are planning to share HTML files with
CSWS (Apache for OpenVMS), it is recommended that you change the
CSWS_JAVA directory tree's ownership to APACHE$WWW.
```

Select Option 1 from the CSWS Jakarta Configuration Menu

Example:

```
Enter configuration option: 1
```

```
Enter the OpenVMS account name for Jakarta (Tomcat) [SYSTEM]: apache$www
```

To operate successfully, the server processes must have read access to the installed files and read-write access to certain other files and directories. It is recommended that you use this procedure to set the owner UIC on the CSWS files and directories to match the server. If you are changing the OpenVMS account name, you might want to change the ownership of the Jakarta tree.

```
Set owner UIC to APACHE$WWW on CSWS java jakarta files (Yes/No) [Yes]: Y
This could take a minute or two . . .
```

After configuration, start CSWS_JAVA entering:

```
$ @SYS$STARTUP:APACHE$JAKARTA_STARTUP
```

Check that neither SYLOGIN.COM nor the LOGIN.COM write any output to SYS\$OUTPUT:.. Look especially for a

```
$ SET TERMINAL/INQUIRE.
```

Start the CSWS_JAVA server at system boot time by adding the following lines to SYS\$MANAGER:SYSTARTUP_VMS.COM:

```
$ file := SYS$STARTUP:APACHE$JAKARTA_STARTUP.COM
$ if f$search("''file'") .nes. "" then @'file'
```

Shutdown the CSWS_JAVA server at system shutdown time by adding the following lines to SYS\$MANAGER:SYSHUTDOWN.COM:

```
$ file := SYS$STARTUP:APACHE$JAKARTA_SHUTDOWN.COM
$ if f$search("''file'") .nes. "" then @'file'
```

Test the installation using your favorite Web browser. Replace host.domain in the following URL (Uniform Resource Locator) with the information for the web server just installed, configured, and started. Note that by default the web server uses port 8080.

URL `http://host.domain:8080/` should display the standard introductory page from the Apache Software Foundation. This has the Tomcat logo in the upper left hand corner.

If you do not see this page, check the CSWS_JAVA release notes.

Thank you for using CSWS_JAVA

Installing CSWS_JAVA on an ODS-5 Enabled Disk

It is necessary to install CSWS_JAVA on an ODS-5 enabled disk because of several Java-related requirements such as case-sensitive filenames, long filename support, and multi-dot filename support. The basic installation of Tomcat 7 ships with several multi-dot filenames.

By default, DCL does not enable extended filename support. To use DCL utilities, you must enable `parse_style=extend` for the process before working with ODS-5 file specifications.

For example, if you enter the following command, you will get an invalid parameter delimiter error, where `2DKB400` is the disk where CSWS_JAVA is installed:

```
$ type $2$DKB400:[APACHE.JAKARTA.TOMCAT.conf]server-noexamples^.xml.config
%DCL-W-PARMDEL, invalid parameter delimiter - check use of special characters
\^\  


```

If you set the process to `parse_style=extend`, the file is displayed properly:

```
$ set proc/parse=extend
$ type $2$DKB400:[APACHE.JAKARTA.TOMCAT.conf]server-noexamples^.xml.config/page
<!-- Alternate Example-less Configuration File -->
<!-- Note that component elements are nested corresponding to their parent-child
relationships with each other -->
.
.
.
```

Using an ODS-5 enabled disk also avoids servlet name space collisions that will occur with the 39.39 character filename limitation on an ODS-2 file system.

Configuring CSWS_JAVA

After the installation is complete, perform the following steps:

1. Change the Jakarta directory tree's ownership to `APACHE$WWW` and set the owner UIC (optional).

The default installation uses the `SYSTEM` account to run the `CSWS_JAVA` Tomcat engine. If you are planning to share HTML files with the Secure Web Server, change the Jakarta directory tree's ownership to `APACHE$WWW` by running the `CSWS_JAVA` configuration utility and selecting the first option.

For example:

```
$ @SYS$STARTUP:APACHE$JAKARTA_CONFIG
Using CATALINA_BASE   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_HOME   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_TMPDIR : /sys$common/apache/jakarta/tomcat/temp
java_vms_base = SYS$COMMON:[JAVA$60.
JAVA_HOME = /SYS$COMMON/JAVA$60
SYSTEM_CLASSPATH = SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR
Using Java 6.0 setup -- $ @sys$manager:java$60_setup
```

```
CSWS Jakarta Configuration Menu
```

```
Configuration Options:
```

- 1 - Change Username
- 2 - Add ACL to Jakarta (Tomcat) directories
- 3 - View current configuration
- 4 - Start CSWS Jakarta (Tomcat) for OpenVMS
- 5 - Stop CSWS Jakarta (Tomcat) for OpenVMS

```
[E]- Exit Configuration procedure
```

```
Enter configuration option: 1
```

```
Enter the OpenVMS account name for Jakarta (Tomcat) [SYSTEM]: apache$www
```

To operate successfully, the server processes must have read access to the installed files and read-write access to certain other files and directories. It is recommended that you use this procedure to set the owner UIC on the CSWS files and directories to match the server. If you are changing the OpenVMS account name, you might want to change the ownership of the Jakarta tree.

```
Set owner UIC to APACHE$WWW on CSWS java jakarta files (Yes/No) [Yes]
```

```
This could take a minute or two . . .
```

```
Update the Jakarta configuration data file (Yes/No) [Yes]
```

Press RETURN to continue

CSWS Jakarta Configuration Menu

Configuration Options:

- 1 - Change Username
- 2 - Add ACL to Jakarta (Tomcat) directories
- 3 - View current configuration
- 4 - Start CSWS Jakarta (Tomcat) for OpenVMS
- 5 - Stop CSWS Jakarta (Tomcat) for OpenVMS

[E]- Exit Configuration procedure

Enter configuration option: E

Important: Check quota requirements for servlet engines.

When you select the user account for Jakarta (Tomcat), consider Java quota requirements to ensure best performance of your Java applications.

In particular, you might need to increase `FILLM` (and the related `CHANNELCNT` `SYSGEN` parameter), `PGFLQUO`, and `BYTLM`.

For more information on Java quota requirements, see the section on Setting Process Quotas for Better Performance on OpenVMS in the Java SDK Release Notes.

2. Ensure that Tomcat is up and running.

If Tomcat is not currently running, start it by entering the following command:

```
$ @SYS$STARTUP:APACHE$JAKARTA_CONFIG
Using CATALINA_BASE   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_HOME   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_TMPDIR : /sys$common/apache/jakarta/tomcat/temp
java_vms_base = SYS$COMMON:[JAVA$60.
JAVA_HOME = /SYS$COMMON/JAVA$60
SYSTEM_CLASSPATH = SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR
Using Java 6.0 setup -- $ @sys$manager:java$60_setup
```

CSWS Jakarta Configuration Menu

Configuration Options:

- 1 - Change Username
- 2 - Add ACL to Jakarta (Tomcat) directories
- 3 - View current configuration
- 4 - Start CSWS Jakarta (Tomcat) for OpenVMS
- 5 - Stop CSWS Jakarta (Tomcat) for OpenVMS

[E]- Exit Configuration procedure

Enter configuration option: 4

```
Using CATALINA_BASE   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_HOME   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_TMPDIR : /sys$common/apache/jakarta/tomcat/temp
java_vms_base = SYS$COMMON:[JAVA$60.
JAVA_HOME = /SYS$COMMON/JAVA$60
SYSTEM_CLASSPATH = SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR
Using Java 6.0 setup -- $ @sys$manager:java$60_setup
Starting Tomcat...
Starting APACHE$TOMCAT as a detached network process
%APACHE-S-PROC_ID, identification of created process is 00007B23
Tomcat Logicals and Classpaths are cleared
Press RETURN to continue
```

3. Optional: Start Tomcat using a different configuration file.

By default, Tomcat uses `CATALINA_HOME/conf/server.xml` for configuration. The default configuration uses `CATALINA_HOME` as its base for the contexts.

You can change this by using the `-f /path/to/server.xml` option, with a different server configuration file and setting the `home` property of the context manager. For more information see, Tomcat 7 Documentation.

Note: On OpenVMS, these commands are case-sensitive. Put quotes around the UNIX portion of the command to retain lower or mixed case.

To change the start-up directory, enter the following:

```
$ @sys$startup:apache$jakarta start "-f" "/path/to/server.xml"
```

4. View the current Tomcat configuration.

Enter the following command and select Option 3. If the Tomcat Servlet engine is running, you will see an `APACHE$TOMCAT` process.

```
$ @SYS$STARTUP:APACHE$JAKARTA_CONFIG
  Using CATALINA_BASE   : /sys$common/apache/jakarta/tomcat/
  Using CATALINA_HOME   : /sys$common/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR : /sys$common/apache/jakarta/tomcat/temp
java_vms_base = SYS$COMMON:[JAVA$60.
JAVA_HOME = /SYS$COMMON/JAVA$60
SYSTEM_CLASSPATH = SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR
Using Java 6.0 setup -- $ @sys$manager:java$60_setup
```

```
CSWS Jakarta Configuration Menu
```

```
Configuration Options:
```

- 1 - Change Username
- 2 - Add ACL to Jakarta (Tomcat) directories
- 3 - View current configuration
- 4 - Start CSWS Jakarta (Tomcat) for OpenVMS
- 5 - Stop CSWS Jakarta (Tomcat) for OpenVMS

```
[E]- Exit Configuration procedure
```

```
Enter configuration option: 3
```

```
  Using CATALINA_BASE   : /sys$common/apache/jakarta/tomcat/
  Using CATALINA_HOME   : /sys$common/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR : /sys$common/apache/jakarta/tomcat/temp
java_vms_base = SYS$COMMON:[JAVA$60.
JAVA_HOME = /SYS$COMMON/JAVA$60
SYSTEM_CLASSPATH = SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR
Using Java 6.0 setup -- $ @sys$manager:java$60_setup
Tomcat environment Initialized
```

```
Jakarta Configuration:
```

```
Configuration file: SYS$MANAGER:APACHE$JAKARTA_CONFIG.DAT
```

```
OpenVMS Account Name: SYSTEM
Tomcat home: /sys$common/apache/jakarta/tomcat/
```

```
Java Version information:
```

```
java version "1.6.0"
Java(TM) SE Runtime Environment "1.6.0-6"
Java HotSpot(TM) 64-Bit Server VM (build 20.6-b06, mixed mode)
```

```
Java$classpath:
"JAVA$CLASSPATH" = "SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR" (LNM$PROCESS_TABLE)
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.bin]bootstrap.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.bin]tomcat-juli.jar"
= "[]"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]annotations-api.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]catalina-ant.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]catalina-ha.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]catalina-tribes.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]catalina.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]ecj-3^7^2.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]el-api.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]jasper-el.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]jasper.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]jsp-api.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]servlet-api.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-api.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-coyote.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-dbcp.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-il8n-es.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-il8n-fr.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-il8n-ja.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-jdbc.jar"
= "SYS$COMMON:[APACHE.JAKARTA.TOMCAT.LIB]tomcat-util.jar"
```

```
Java$filename_controls:
```

```
"JAVA$FILENAME_CONTROLS" = "8" (LNM$PROCESS_TABLE)
"JAVA$FILENAME_CONTROLS" = "-1" (LNM$JOB_88D3DF80)
```

```
show sys/m/owner=APACHE$WWW :
```

```
OpenVMS 8.4-1H1 on node EXEC14 1-DEC-2015 16:29:27.21 Uptime 7 12:54:27
Pid Process Name State Pri I/O CPU Page flts Pages
00007B23 APACHE$TOMCAT HIB 6 14634 0 00:00:18.43 17717 17290 M
Press RETURN to continue
```

Note: The first invocation of Tomcat completes the installation of the environment, so there is a delay before Tomcat is ready to serve JSP pages. Subsequent invocations of Tomcat will be faster.

5. If the Tomcat servlet engine does not start, check the log files in the default directory of the account.

Enter the following commands:

```
$ DIR APACHE$ROOT:[000000]APACHE$JAKARTA*.LOG
Directory APACHE$ROOT:[000000]
APACHE$JAKARTA_SERVER_OUTPUT.LOG;1
Total of 1 file.
$ TYPE APACHE$ROOT:[000000]APACHE$JAKARTA_SERVER_OUTPUT.LOG
$ Set Noon
$ VERIFY = F$VERIFY(F$TRNLNM("SYLOGIN_VERIFY"))
Using CATALINA_BASE : /sys$common/apache/jakarta/tomcat/
Using CATALINA_HOME : /sys$common/apache/jakarta/tomcat/
Using CATALINA_TMPDIR: /sys$common/apache/jakarta/tomcat/temp
/sys$common/apache/jakarta/tomcat/temp
```

```

java_vms_base = SYS$COMMON:[JAVA$60.JAVA_HOME = /SYS$COMMON/JAVA$60
SYSTEM_CLASSPATH = SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR
Using Java 6.0 setup - $ @sys$manager:java$60_setup
Running Tomcat.....
Exceeded quota, Please raise paging file quota
Requires a minimum of 200,000 free
Current available is: 100000
%SYSTEM-F-EXQUOTA, process quota exceeded
  APACHE$WWW      job terminated at 1-DEC-2015 16:13:32.31
Accounting information:
Accounting information:
Buffered I/O count:           81      Peak working set size:      2016
Direct I/O count:            74      Peak virtual size:         167933
Page faults:                 58      Mounted volumes:           0
Charged CPU time:           0 00:00:00.10  Elapsed time:             0 00:00:00.29

```

6. Access the included JSP and servlet examples via <http://hostname:8080> after you have successfully configured and started Tomcat (replacing hostname with the name or IP address of the server on which Tomcat has been installed).

Optional Settings

The following optional settings require a `.TOMCATRC` file. This file must be in the `SYS$LOGIN` directory of the OpenVMS account used for Tomcat, typically the `APACHE$WWW` account. You can determine the account by viewing the current configuration shown described elsewhere in this document.

If you are running Tomcat from the `APACHE$WWW` account and the `.TOMCATRC` file does not exist, you must create it. For example:

```

$ CREATE APACHE$ROOT:[000000].TOMCATRC
^Z
$

```

1. **Optional: Add new `CLASSPATH` entries.**

To add new `CLASSPATH` entries (for example, JDBC drivers), add the following line to your `.TOMCATRC` file:

```

$ DEFINE APACHE$JAKARTA_USER_CLASSPATH NAVROOT:[JAVA]NVJDBC1.JAR

```

2. **Optional: Supply additional JVM command line parameters.**

You may need to supply additional JVM command line parameters if, for example, you need to increase the maximum heap size to 128MB.

```

$ CREATE TOMCAT_JVM_ARGS.DAT
-mx128m
^Z

```

3. **Add the following line to your `.TOMCATRC` file:**

```

$ def APACHE$JAKARTA_JAVA_PARAMETERS_FILE disk:[directory]TOMCAT_JVM_ARGS.DAT

```

Note: Make sure that the `APACHE$WWW` account can read these files.

4. **Optional: Override `JAVA$FILENAME_CONTROLS` default.**

To override the default `JAVA$FILENAME_CONTROLS` logical name value (8) set by the configuration procedure, add the following line to your `.TOMCATRC` file, where `n` is the value that should be assigned to the `JAVA$FILENAME_CONTROLS` logical name.

```
$ DEFINE APACHE$JAKARTA_FILENAME_CONTROLS n
```

The Release Notes for the Java Software Development Kit (SDK) describes the `JAVA$FILENAME_CONTROLS` logical name and how it can be used to reduce filename mappings and improve performance with ODS-5 disks.

By default, the `CSWS_JAVA` configuration will set `JAVA$FILENAME_CONTROLS` to 8, which allows mixed UNIX/VMS-style filenames, overriding the original value of -1 (all mappings, to support ODS-2, lower performance) set by the Java setup procedure:

```
"JAVA$FILENAME_CONTROLS" = "8" (LNM$PROCESS_TABLE)
"JAVA$FILENAME_CONTROLS" = "-1" (LNM$JOB_8165E800)
```

If you define `APACHE$JAKARTA_FILENAME_CONTROLS` in `.TOMCATRC`, the `CSWS_JAVA` configuration will use that value to override the default. For example, if you put the following lines in `.TOMCATRC`:

```
$ FILE_MASK = %x00000008 + %x00000200
$ DEFINE JAVA$FILENAME_CONTROLS 'file_mask'
```

the `CSWS_JAVA` configuration procedure will set `JAVA$FILENAME_CONTROLS` as follows:

```
"JAVA$FILENAME_CONTROLS" = "520" (LNM$PROCESS_TABLE)
"JAVA$FILENAME_CONTROLS" = "-1" (LNM$JOB_8165E800)
```

This setting allows mixed UNIX/OpenVMS-style filenames and `.DIR` in filenames.

Building the Sample Web Application on OpenVMS

To build the sample web application, perform the following steps:

- Set your directory to the sample directory:

```
$ SET DEF apache$common:[JAKARTA.TOMCAT.webapps.docs.appdev.sample]
```

- Enter the following build command:

```
$ @SYS$STARTUP:APACHE$JAKARTA ANT "-buildfile" build.xml "dist" -
"-Dcatalina.home=/apache$common/jakarta/tomcat"
```

You will then see output similar to the following:

```
Using CATALINA_BASE   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_HOME   : /sys$common/apache/jakarta/tomcat/
Using CATALINA_TMPDIR : /sys$common/apache/jakarta/tomcat/temp
java_vms_base = SYS$COMMON:[JAVA$60.
JAVA_HOME = /SYS$COMMON/JAVA$60
SYSTEM_CLASSPATH = SYS$COMMON:[JAVA$60.LIB]TOOLS.JAR
Using Java 6.0 setup -- $ @sys$manager:java$60_setup
Run ANT in Tomcat's environment
Buildfile: BUILD.XML

prepare:
[mkdir] Created dir:
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/build
[mkdir] Created dir:
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/build/WEB-INF
```

```

    [mkdir] Created dir:
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/build/WEB-INF/classes
    [copy] Copying 4 files to
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/build
    [mkdir] Created dir:
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/build/WEB-INF/lib

compile:
    [javac] Compiling 1 source file to
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/build/WEB-INF/classes

javadoc:
    [mkdir] Created dir:
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/dist/docs/api
    [javadoc] Generating Javadoc
    [javadoc] Javadoc execution
    [javadoc] Loading source files for package mypackage...
    [javadoc] Constructing Javadoc information...
    [javadoc] Standard Doclet version 1.6.0
    [javadoc] Building tree for all the packages and classes...
    [javadoc] Building index for all the packages and classes...
    [javadoc] Building index for all classes...

dist:
    [copy] Copying 1 file to
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/dist/docs
    [jar] Building jar:
/apache$root/jakarta/tomcat/webapps/docs/appdev/sample/dist/myapp-0.1-dev.war

BUILD SUCCESSFUL
Total time: 9 seconds
Tomcat Logicals and Classpaths are cleared

```

- Copy the .WAR file to the webapps directory. For example:

```

$ COPY NODE11$DKA0:[APACHE.JAKARTA.TOMCAT.webapps.docs.appdev.sample.dist]myapp-
0^1-dev.war NODE11$DKA0:[APACHE.JAKARTA.TOMCAT.webapps]myapp.war

```

- Stop Tomcat (if it is running) by entering:

```

$ @SYSS$STARTUP:APACHE$JAKARTA

```

Select Option 5, Stop CSWS Jakarta (Tomcat) for OpenVMS.

- Start Tomcat (if it is running) by entering:

```

$ @SYSS$STARTUP:APACHE$JAKARTA

```

Selecting Option 4, Start CSWS Jakarta (Tomcat) for OpenVMS.

- Enter the following URL to access the sample application:

<http://hostname:8080/myapp/index.html>

You should see a page with links to a JSP or servlet file. Selecting either page produces a display of the request headers.

Running Tomcat

For information about running Tomcat, see Tomcat 7 Documentation.

Release Notes

This section contains notes about the Tomcat component of the current release of CSWS_JAVA.

- 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. To avoid this slow startup, delete the sub directories in the webapps directory.

- Configuration dialog question about updating configuration data file

When you run `APACHE$JAKARTA` or `APACHE$JAKARTA_CONFIG`, you are prompted with this question "Update the Jakarta configuration data file? (Yes/No) [Yes]".

Enter Yes if you want the new changes to be reflected in the configuration file (`APACHE$JAKARTA_CONFIG.DAT`). In future, you can have a development Tomcat server and a production Tomcat server on the same system, but with different configuration information for each server.

- Redeploying .WAR file fails

Previously, to redeploy .WAR files you had to delete all the 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 login.com file of the default directory where Tomcat is executed (or within the .tomcatrc file under the same directory) and restart Tomcat:

```
$ define java$delete_all_versions 1
$ define java$create_dir_with_owner_delete 1
```

- System without Motif installed displays an error during Tomcat start-up

If you start Tomcat on a system that does not have Motif installed (such as on a "headless workstation"), the error message "Java.lang.UnsatisfiedLinkError: no such file or directory" is displayed.

To avoid the message getting displayed during start-up, do the following:

- Create a `headless_data.dat` file in the `apache$common:[000000]` directory:

```
$ create headless_data.dat
Then add the following commands to the file:
-Djava.awt.headless=true
-Djava.awt.headlesslib=true
^Z
```

- Add the following lines in the `apache$root:login.com` file:

```
$ def APACHE$JAKARTA_JAVA_PARAMETERS_FILE -
apache$common:[000000]headless_data.dat
```

- Start Tomcat.