CSWS_JAVA for HP Secure Web Server for OpenVMS (based on Apache)

Installation Guide and Release Notes

November 2005

CSWS_JAVA Version 3.0 for OpenVMS Alpha (based on Apache Tomcat 5.5.9)
CPQ-AXPVMS-CSWS_JAVA-V0300--1.PCSI_SFX_AXPEXE

CSWS_JAVA Version 3.0 for OpenVMS I64 (based on Apache Tomcat 5.5.9)
HP-I64VMS-CSWS_JAVA-V0300--1.PCSI_SFX_I64EXE

Contents

What's New
Overview
Software Prerequisites
Documentation
Downloading the Kit
Expanding the Kit
Before Beginning the Installation
Installing CSWS_JAVA
    Installing CSWS_JAVA on an ODS-5 Enabled Disk
Configuring CSWS_JAVA
    Optional Settings
Building the Sample Web Application on OpenVMS
Running Tomcat
Release Notes

What's New

CSWS_JAVA is available on OpenVMS Alpha and OpenVMS I64. Version 3.0 is intended to work with the **Secure Web Server Versions 2.1** and **1.3-1**. It does not work with SWS V2.0.

CSWS_JAVA Version 3.0 contains support for **Tomcat 5.5.9**. The previous version of CSWS_JAVA, Version 2.1, contained support for Tomcat 4.1.24.

Apache Tomcat 5.5.x is the current focus of development at jakarta.apache.org. While it supports the same servlet and JSP specification versions as Apache Tomcat 5.0.x, there are significant changes in many areas, resulting in improved performance, stability, and total cost of ownership.

Apache Tomcat 5.0.x improves on Apache Tomcat 4.1 in the following ways:

- Performance optimizations and reduced garbage collection
- Refactored application deployer, with an optional standalone deployer allowing validation and compilation of a web application before it goes into production
- Complete server monitoring using JMX and the manager web application

- Scalability and reliability enhancements
- Improved Taglibs handling, including advanced pooling and tag plugins
- Improved platform integration, with native Windows and Unix wrappers
- Embedding using JMX
- Enhanced Security Manager support
- Integrated session clustering
- Expanded documentation

Overview

CSWS_JAVA includes the following projects:

- Tomcat (Catalina)
- Ant
- Jspc

See the Jakarta Apache Project for more information about Tomcat and other projects.

Tomcat

Tomcat is an extension to the HP Secure Web Server, but it runs independently of SWS in a separate process. You can configure your system so that the Secure Web Server serves HTML pages, while Tomcat serves the JSP pages and runs the servlets.

CSWS_JAVA includes the following Apache Tomcat technologies:

- JavaServer Pages 2.0
- Java Servlet 2.4
- MOD_JK

Tomcat is the reference implementation for the Java Servlet 2.4 and JavaServer Pages 2.0 technologies. CSWS_JAVA includes the final Tomcat Version 5.5.9.

Tomcat is a servlet container with a JSP environment. A servlet container is a runtime shell that manages and invokes 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.

MOD_JK is an optimized version of the HTTP protocol that allows a standalone web server such as Apache to talk to Tomcat. MOD_JK2 is a refactoring of MOD_JK and uses the Apache Portable Runtime (apr).
Note: The Apache Jakarta project currently lists MOD_JK2 as unsupported.

For more information about MOD_JK and MOD_JK2, see What's JK?

See Tomcat 5.5 Documentation for more information.

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.

Apache JServ

Beginning with CSWS_JAVA V2.0, support has been retired for Apache JServ, which was provided in the CSWS_JSERV kit. If you want to continue using Apache JServ, download CSWS_JAVA V1.1.

Software Prerequisites

CSWS_JAVA for the *Secure Web Server for OpenVMS* requires the following software:

- OpenVMS Alpha Version 7.3-2 (or higher)
  – **or** –
  OpenVMS I64 Version 8.2 (or higher)
  (Check for the latest required patches from http://www1.itrc.hp.com/service/patch/mainPage.do)

- Secure Web Server Version 1.3-1 or 2.1 (or higher) for OpenVMS
  (CSWS Version 1.3-1 or 2.1 (or higher) is required to run Tomcat with the Secure Web Server. Tomcat can also be used by itself, without the Secure Web Server.)

- On OpenVMS Alpha: Software Development Kit (SDK) for the OpenVMS Operating System, for the Java™ Platform v 1.4.2

- On OpenVMS I64: Software Development Kit (SDK) for the OpenVMS Operating System, for the Java™ Platform v 1.4.2

- All SDK 1.4.2 patches required for your version of OpenVMS

- HP **requires** that you install CSWS_JAVA on an ODS-5 enabled disk. Your installation of the Secure Web Server can remain on an ODS-2 disk.

Documentation

For information about Tomcat, see the Jakarta Apache Project and Tomcat 5.5 Documentation. General information about Apache is available from the Apache Software Foundation.

Downloading the Kit

» **Download CSWS_JAVA Version 3.0 for HP Secure Web Server for OpenVMS Alpha**
» **Download CSWS_JAVA Version 3.0 for HP Secure Web Server for OpenVMS I64**

Also available for download are earlier versions of CSWS_JAVA:

» **Download CSWS_JAVA Version 2.1 for HP Secure Web Server for OpenVMS Alpha**
» **Download CSWS_JAVA Version 2.1 for HP Secure Web Server for OpenVMS I64**

If you download Version 2.1, see the CSWS_JAVA Version 2.1 Installation Guide and Release Notes.

Expanding the Kit

To expand the CSWS_JAVA self-extracting file, enter one of the following commands, depending on the kit you download:

```
$ RUN CPQ-AXPVMS-CSWS_JAVA-V0300--1.PCSI_SFX_AXPEXE  ! on Alpha
$ RUN HP-I64VMS-CSWS_JAVA-V0300--1.PCSI_SFX_I64EXE   ! on I64
```

The system expands the file and names the decompressed file CPQ-AXPVMS-CSWS_JAVA-V0300--1.PCSI or HP-I64VMS-CSWS_JAVA-V0300--1.PCSI. Do not rename this file.

Before Beginning the Installation

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

1. **Shut down the Secure Web Server**.

   ```
   $ @SYS$STARTUP:APACHE$SHUTDOWN
   ```

2. **Shut down Tomcat**.

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

3. **Remove CSWS_JSERV, if installed.**

   Before you install CSWS_JAVA, HP recommends that you remove CSWS_JSERV, if it was previously installed.

   Perform a backup of any user files contained in the [APACHE.JSERV] directory tree, then enter the following commands to remove CSWS_JSERV:

   ```
   $ PRODUCT REMOVE CSWS_JSERV
   ```

4. **Delete the JSERV startup command procedure.**

   Enter the following command:

   ```
   $ DELETE APACHE$ROOT:[000000]START_JSERV_MANUAL.COM;*
   ```

5. **Remove previous version of CSWS_JAVA, if installed.**

   Before installing CSWS_JAVA Version 3.0, you must **manually remove the previous version** of CSWS_JAVA if it is installed on your system. This will remove the CSWS_JAVA dependencies in the Secure Web Server. By removing the dependencies first, installing CSWS_JAVA Version 3.0 will not inadvertently bring down the Secure Web Server.

   To remove the previous version of CSWS_JAVA, perform the following steps:

   - Disable any MOD_JK or MOD_JK2 adapters that were configured for the Secure Web Server by entering:

     ```
     $ @SYS$STARTUP:APACHE$JAKARTA
     ```

Select Configure Apache's httpd.conf for Jakarta Adapters.
Then select Disable mod_jk, Disable mod_jk2, or Disable mod_jk2 (Apache 2.1).

Important: Perform a backup of any user files in the [apache.jakarta] directory tree.

- 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

HP **requires** that you install CSWS_JAVA on an ODS-5 enabled disk. Your installation of the Secure Web Server can remain on an ODS-2 disk. **You do not need to install CSWS_JAVA into the same disk or directory as the Secure Web Server.**

1. **Verify that the destination device is an ODS-5 volume** by entering a command similar to the following, where *DISK$DKA200* is the disk where you want to install CSWS_JAVA:

```
$ SHOW DEV DISK$DKA200/FULL

Disk VARMIT$DKA200:, device type BB00923468, is online, mounted, file-
oriented device, shareable, available to cluster, error logging is enabled.
                            .
                            .

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

2. **Install the CSWS_JAVA kit** by entering the following command, where *DISK$DKA200* is the name of the ODS-5 enabled disk where you want to install CSWS_JAVA. **Be sure that you manually removed the previous version of CSWS_JAVA before proceeding**.

```
$ PRODUCT INSTALL CSWS_JAVA/DEST=DISK$DKA0:[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 on an Alpha system. A similar log is displayed on an I64 system.

```
$ PRODUCT INSTALL CSWS_JAVA/DEST=DISK$DKA0:[000000]

The following product has been selected:
    CPQ AXPVMS CSWS_JAVA V3.0              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.
```

```
CPQ AXPVMS CSWS_JAVA V3.0

    Hewlett-Packard Company & The Apache Software Foundation.

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
    CPQ AXPVMS CSWS_JAVA V3.0             DISK$RAMDISK:[000000.]

Portion done: 0%...10%...20%...30%...40%...50%...60%...90%...100%

The following product has been installed:
    CPQ AXPVMS CSWS_JAVA V3.0             Layered Product

CPQ AXPVMS CSWS_JAVA V3.0

Post installation tasks required for CSWS_JAVA for OpenVMS Alpha

    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 HP's Secure Web Server, it is
recommended that you change the Jakarta 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.  HP recommends 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 (Jakarta) by
    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 (Jakarta) 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 (Jakarta) server at system shutdown time by adding the following
lines to SYS$MANAGER:SYSHUTDWN.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 HP Secure Web Server just installed, configured, and started.

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.
```

3. When the installation completes, **start the Secure Web Server** by entering the following command:

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

   If the web server does not restart, check APACHE$ROOT:[000000]APACHE$$SERVER.LOG for errors.

Installing CSWS_JAVA on an ODS-5 Enabled Disk

HP requires that you install CSWS_JAVA Version 3.0 on an ODS-5 enabled disk because of several issues: case sensitive filenames, long filename support, and multidot filename support. The basic installation of Tomcat 5.5 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 get an invalid parameter delimiter error:

```
$ type [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 [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 collision that will occur with the 39.39 character filename limitation on an ODS-2 file system.

For example, the URL http://hostname/examples/jsp/num/numguess.jsp builds a temporary file in the directory [APACHE.JAKARTA.WORK.localhost_8080_2Fexamples] called

```
_0002fjsp_0002fnum_0002fnumguess_0002ejspnumguess_jsp_0.java
```

On an ODS-2 file system, the filename is truncated to

```
_0002FJSP_0002FNUM_0002FNUMGUESS_0002EJ.JAVA
```

If your URL grows to http://hostname/examples/jsp/num/numguess.jsp/new_numguess.jsp the uniqueness of the filename on an ODS-2 file system comes into question, and you will start to see internal servlet errors (wrong name errors).

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.**

   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  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_HOME  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR: /secure$mda0/apache/jakarta/tomcat/temp
  Using Java 1.4.0 setup

     CSWS Jakarta Configuration Menu

     Configuration Options:

          1 - Change Username
          2 - Add ACL to Jakarta (Tomcat) directories
          3 - Configure Apache's httpd.conf for Jakarta Adapters
          4 - View current configuration
          5 - Start CSWS Jakarta (Tomcat) for OpenVMS
          6 - 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. HP
recommends 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 - Configure Apache's httpd.conf for Jakarta Adapters
          4 - View current configuration
          5 - Start CSWS Jakarta (Tomcat) for OpenVMS
          6 - 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 the Jakarta (Tomcat) or JServ servlet engines, consider Java quota requirements to ensure best performance of your Java applications.

The default quota values for the APACHE$WWW account that are set by the *Secure Web Server* installation might not be optimized for Java. In particular, you might need to increase FILLM (and the related CHANNELCNT SYSGEN parameter), PGFLQUO, and BYTLM. These are pooled quotas. If you are configuring the JServ servlet engine, which is a subprocess, you need to be aware of the impact on these quotas from other Apache child processes in the same job tree. The Jakarta (Tomcat) servlet engine is a detached process and is not affected by Apache child processes.

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

2. **Configure MOD_JK2 support.**

   For example:

```
   $ @SYS$STARTUP:APACHE$JAKARTA_CONFIG

  Using CATALINA_BASE  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_HOME  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR: /secure$mda0/apache/jakarta/tomcat/temp

  Using Java x.x.x setup

     CSWS Jakarta Configuration Menu

     Configuration Options:

          1 - Change Username
          2 - Add ACL to Jakarta (Tomcat) directories
          3 - Configure Apache's httpd.conf for Jakarta Adapters
          4 - View current configuration
          5 - Start CSWS Jakarta (Tomcat) for OpenVMS
          6 - Stop  CSWS Jakarta (Tomcat) for OpenVMS
```

```
         [E]- Exit Configuration procedure

Enter configuration option: 3

     CSWS Jakarta Adapter Configuration Menu

     Configuration Options:

          1 - Enable mod_jk                 httpd.conf
          2 - Disable mod_jk                httpd.conf
          3 - Enable mod_jk2                httpd.conf ! for SWS 1.3-1
          4 - Disable mod_jk2               httpd.conf
          5 - Enable mod_jk2    (Apache 2.1) httpd.conf ! for SWS 2.1
          6 - Disable mod_jk2   (Apache 2.1) httpd.conf
          7 - Enable mod_webapp (Apache 2.1) httpd.conf
          8 - Disable mod_webapp(Apache 2.1) httpd.conf
          9 - Restart CSWS (Apache) for OpenVMS

         [E]- Exit Configuration procedure

Enter configuration option: 3
Location of httpd.conf [APACHE$COMMON:[CONF]HTTPD.CONF]
Mod_jk2 configuration file
[SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.CONF.JK2]MOD_JK2.CONF]

copying the default workers2.properties file...
Copy [apache.jakarta.tomcat.conf.jk2]workers2_def.properties
apache$common:[conf]workers2.properties

        Before restarting CSWS please ensure that Tomcat is
        up and running.  (Test page http://hostname:8080/)

        Failure to start Tomcat before CSWS could result in a
        failure to load the mod_jk2 module during CSWS startup
```

3. **Ensure that Tomcat is up and running.**

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

```
   $ @SYS$STARTUP:APACHE$JAKARTA

     CSWS Jakarta Configuration Menu

     Configuration Options:

          1 - Change Username
          2 - Add ACL to Jakarta (Tomcat) directories
          3 - Configure Apache's httpd.conf for Jakarta Adapters
          4 - View current configuration
          5 - Start CSWS Jakarta (Tomcat) for OpenVMS
          6 - Stop  CSWS Jakarta (Tomcat) for OpenVMS

         [E]- Exit Configuration procedure

  Enter configuration option: 5
```

```
  Using CATALINA_BASE  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_HOME  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR: /secure$mda0/apache/jakarta/tomcat/temp
Using Java x.x.x setup
Setting up symbols for foreign command line usage...
JAVA$FILENAME_CONTROLS now set to: -1
Starting Tomcat...
Starting APACHE$TOMCAT as a detached network process
%APACHE-S-PROC_ID, identification of created process is 00000236
Tomcat Logicals and Classpaths are cleared
Press RETURN to continue
```

Be sure you can access the test page at http://hostname:8080/

Failure to start Tomcat before restarting Secure Web Server could result in a failure to load the MOD_JK2 module during Secure Web Server startup.

4. **If the Secure Web Server is currently running, restart it so that these configuration changes take effect.**

   To restart the Secure Web Server, enter the following command:

   ```
   $ @SYS$STARTUP:APACHE$STARTUP RESTART
   ```

5. **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. See Tomcat 5.5 Documentation for more information.

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

   To change the startup directory, enter the following:

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

6. **View the current Tomcat configuration.**

   Enter the following command and select Option 4. If the Tomcat Servlet engine is running, you will see a **APACHE$TOMCAT** process.

   ```
   $ @SYS$STARTUP:APACHE$JAKARTA

     CSWS Jakarta Configuration Menu

     Configuration Options:

          1 - Change Username
          2 - Add ACL to Jakarta (Tomcat) directories
   ```

```
          3 - Configure Apache's httpd.conf for Jakarta Adapters
          4 - View current configuration
          5 - Start CSWS Jakarta (Tomcat) for OpenVMS
          6 - Stop  CSWS Jakarta (Tomcat) for OpenVMS

         [E]- Exit Configuration procedure

Enter configuration option: 4

  Using CATALINA_BASE  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_HOME  : /secure$mda0/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR: /secure$mda0/apache/jakarta/tomcat/temp
Using Java x.x.x setup
Tomcat  environment Initialized

Jakarta Configuration:


 Configuration file: SYS$COMMON:[SYSMGR]APACHE$JAKARTA_CONFIG.DAT

   OpenVMS Account Name:     APACHE$WWW
   Tomcat home:              /secure$mda0/apache/jakarta/tomcat/
   MOD_JK2:                   ENABLED
      Include line: SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.CONF.JK2]MOD_JK2.CONF

   Java Version information:

java version "x.x.x"
Java(TM) 2 Runtime Environment, Standard Edition
Classic VM (build x.x.x-x, xx/xx/200x-10:51, native threads, jit)


   Java$classpath:
   "JAVA$CLASSPATH" = "SYS$COMMON:[JAVA$xxx.LIB]TOOLS.JAR" (LNM$PROCESS_TABLE)
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.bin]bootstrap.jar"
        = "[]"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]activation.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]ant.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]commons-collections.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]commons-dbcp.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]commons-logging-api.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]commons-pool.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]jasper-compiler.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]jasper-runtime.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]jdbc2_0-stdext.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]jndi.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]jta.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]mail.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]naming-common.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]naming-factory.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]naming-resources.jar"
        = "SECURE$MDA0:[APACHE.JAKARTA.TOMCAT.common.lib]servlet.jar"

Java$filename_controls:

   "JAVA$FILENAME_CONTROLS" = "8" (LNM$PROCESS_TABLE)
   "JAVA$FILENAME_CONTROLS" = "-1" (LNM$JOB_8181DE00)

   show sys/m/owner=APACHE$WWW :

OpenVMS V7.3-2 on node SECURE   1-MAY-2005 09:46:05.01  Uptime  6 22:27:46
  Pid    Process Name    State  Pri      I/O       CPU       Page flts  Pages
00000555 APACHE$TOMCAT   HIB      4   134422   0 00:04:08.75     14798   6283 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.

7. **If the Secure Web Server Jakarta 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"))
-> Tomcat Directory /SYS$COMMON/APACHE/JAKARTA/
Using Java x.x.x setup
Setting up symbols for foreign command line usage...
JAVA$FILENAME_CONTROLS now set to: -1
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
  SYSTEM       job terminated at  1-AUG-2005 09:31:53.02

  Accounting information:
  Buffered I/O count:  81  Peak working set size:  2016
  Direct I/O count:    74  Peak virtual size:      167936
  Page faults:         58  Mounted volumes:        0
  Charged CPU time:  0 00:00:00.10
  Elapsed time:   0 00:00:00.24
```

8. **Access the included jsp and servlet examples** via http://hostname:8080 after you have successfully configured and started Tomcat. If you have configured the *Secure Web Server* to work with Tomcat, you also can access the examples with the *Secure Web Server* via http://hostname/examples.

   See the Building the Sample Web Application on OpenVMS for instructions on how to run the sample application.

Optional Settings

The following optional settings require a .TOMCATRC file. This file must be in the SYS$LOGIN directory of the OpenVMS account for JAKARTA, typically the APACHE$WWW account. You can determine the account by viewing the current configuration shown in Step 6.

If you are running Tomcat from APACHE$WWW 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 128 MB. (Use the appropriate command line settings for the version of Java that is installed. For more information, enter java -h)

   Create a text file with all of the JVM command line settings. Enter the following command:

   ```
   $ CREATE TOMCAT_JVM_ARGS.DAT
   -mx128m
   ^Z
   ```

   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.

3. **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:

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

   where *n* is the value that should be assigned to the JAVA$FILENAME_CONTROLS logical name.

   The Release Notes for the 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 found in `[APACHE.JAKARTA.TOMCAT.webapps.tomcat-docs.appdev.sample]`, perform the following steps.

1. **Set your directory to the sample directory.**

```
$ SET DEFAULT DKB600:[APACHE.JAKARTA.TOMCAT.webapps.tomcat-docs.appdev.sample]
```

2. **Enter the following build command**, where *dkb600* is the disk where you installed CSWS_JAVA.

```
$ @SYS$STARTUP:APACHE$JAKARTA ANT "-buildfile" build.xml -
_$ "dist" "-Dcatalina.home=/dkb600/apache/jakarta/tomcat"
```

You will then see the following output:

```
  Using CATALINA_BASE  : /secure$dkb600/apache/jakarta/tomcat/
  Using CATALINA_HOME  : /secure$dkb600/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR: /secure$dkb600/apache/jakarta/tomcat/temp

Using Java x.x.x setup
Run ANT in Tomcat's environment
Buildfile: BUILD.XML

prepare:
    [mkdir] Created dir: /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/build
    [mkdir] Created dir: /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/build/WEB-INF
    [mkdir] Created dir: /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/build/WEB-INF/classes
     [copy] Copying 4 files to /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/build
    [mkdir] Created dir: /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/build/WEB-INF/lib

compile:
    [javac] Compiling 1 source file to /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/build/WEB-INF/classes


javadoc:
    [mkdir] Created dir: /dkb600/apache/jakarta/tomcat/webapps/tomcat-

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 x.x.x

  [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 /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/dist/docs
      [jar] Building jar: /dkb600/apache/jakarta/tomcat/webapps/tomcat-
docs/appdev/sample/dist/myapp-0.1-dev.war

BUILD SUCCESSFUL
Total time: 15 seconds
Tomcat Logicals and Classpaths are cleared
```

3. **Copy the `war` file to the `webapps` directory. For example:**

   ```
   $ COPY DKB600:[APACHE.JAKARTA.TOMCAT.webapps.tomcat-
   docs.appdev.sample.dist]myapp-0^.1-dev.war -
   _$ DKB600:[APACHE.JAKARTA.TOMCAT.webapps]myapp.war
   ```

4. **Stop Tomcat (if it is running) by entering:**

   ```
   $ @SYS$STARTUP:APACHE$JAKARTA
   ```

   and selecting Option 6, Stop CSWS Jakarta (Tomcat) for OpenVMS.

5. **Start Tomcat by entering:**

   ```
   $ @SYS$STARTUP:APACHE$JAKARTA
   ```

   and selecting Option 5, Start CSWS Jakarta (Tomcat) for OpenVMS.

6. **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.

7. **Optional: Access the sample application through the Secure Web Server via the MOD_JK2 adapter:**

   Using the MOD_JK2 adapter, add the following lines to APACHE$ROOT:[CONF]WORKERS2.PROPERTIES as follows:

   ```
   # myapps Uri mapping
   [uri:/myapp/*]
   group=lb
   ```

8. **Restart the Secure Web Server**:

   ```
   $ @SYS$STARTUP:APACHE$STARTUP RESTART
   ```

Enter the following URL to access the sample application through the Secure Web Server:

```
http://hostname/myapp/index.html
```

Running Tomcat

For information about running Tomcat, see Tomcat 5.5 Documentation.

Release Notes

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

- Support for Apache JServ retired

  Beginning with CSWS_JAVA V2.0, support for Apache JServ has been retired. This support was provided in the CSWS_JSERV kit. If you want to continue using Apache JServ, download CSWS_JAVA Version 1.1.

- Setting up Tomcat to use Fast VM (Alpha only)

  If you want to use Fast VM with Tomcat, download and install the Fast VM for Java kit from http://h18012.www1.hp.com/java/download/index.html.

  Then define the following logical in the .TOMCATRC file:

```
$ define APACHE$JAKARTA_USE_FASTVM true
```

- Using the -SECURITY parameter prevents the APACHE$TOMCAT process from starting

  Starting CSWS_JAVA Version 3.0 using the "-SECURITY" parameter prevents the APACHE$TOMCAT process from starting.

  This problem will be corrected in a future release of CSWS_JAVA.

  An example of the output when you use the –SECURITY parameter is as follows:

```
$ @SYS$STARTUP:APACHE$JAKARTA START –SECURITY
$ TYPE APACHE$ROOT:[000000]APACHE$JAKARTA_SERVER_OUTPUT.LOG
$ Set NoOn
$ VERIFY = F$VERIFY(F$TRNLNM("SYLOGIN_VERIFY"))
  Using CATALINA_BASE  : /secure$dkb400/apache/jakarta/tomcat/
  Using CATALINA_HOME  : /secure$dkb400/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR: /secure$dkb400/apache/jakarta/tomcat/temp
Using Java 1.4.2 setup
Running Tomcat.....
...with Security Manager
This release of Apache Tomcat was packaged to run on J2SE 5.0 or later. It can
be run on earlier JVMs by downloading and installing a compatibility package
from the Apache Tomcat binary download page. Tomcat Logicals and Classpaths are
cleared.
```

- Building the sample application incurs a "%DCL-W-NOCOMD" warning

  In CSWS_JAVA Version 3.0, building the sample application incurs a %DCL-W-NOCOMD warning. This problem will be corrected in a future release of CSWS_JAVA.

```
$ SET DEFAULT DKB400:[APACHE.JAKARTA.TOMCAT.webapps.tomcat-docs.appdev.sample]
$ @SYS$STARTUP:APACHE$JAKARTA ANT "-buildfile" build.xml "dist" -
_$ "-Dcatalina.home=/dkb400/apache/jakarta/tomcat"
  Using CATALINA_BASE  : /secure$dkb400/apache/jakarta/tomcat/
  Using CATALINA_HOME  : /secure$dkb400/apache/jakarta/tomcat/
  Using CATALINA_TMPDIR: /secure$dkb400/apache/jakarta/tomcat/temp
  Using Java 1.4.2 setup
Run ANT  in Tomcat's environment
Buildfile: BUILD.XML

prepare:

compile:

javadoc:
  [javadoc] Generating Javadoc
  [javadoc] Javadoc execution
  [javadoc] %DCL-W-NOCOMD, no command on line - reenter with alphabetic first
character
  [javadoc] %DCL-W-NOCOMD, no command on line - reenter with alphabetic first
character

dist:

BUILD SUCCESSFUL
Total time: 13 seconds
Tomcat Logicals and Classpaths are cleared
$
```

- Error during initial creation of shared memory file

  If the mod_jk2 (Apache 2.0) adapter is enabled for CSWS_JAVA Version 3.0, an error can occur during the initial creation of the shared memory file (shm.file). The error_log will report the following error:

```
[Thu Jan 15 09:38:50 2004] [error] shm.create(): error creating
/apache$root/logs/shm.file 0 22 0x2e0c30 invalid argument
[Thu Jan 15 09:38:50 2004] [error] shm.create(): error mmapping
/apache$root/logs/shm.file
```

  If this error occurs, restart both CSWS and CSWS_JAVA using supported command procedures. Creation of the shared memory file will be completed during the first shutdown of CSWS. Do not delete the shm.file between restarts.

  This problem will be corrected in a future release of CSWS_JAVA.

- Slow access the first time Tomcat server is invoked

  The first time you invoke the Tomcat server, several minutes may pass before you can access http://hostname:8080. The reason for this is that Tomcat deploys all of the applications (mostly examples) in

the webapps directory. This is only done the first time the server is invoked. If you delete the subdirectories in the webapps directory, you can avoid the slow first time startup.

- Configuration dialog question about updating configuration data file

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

  This question is asking whether you want the new changes to be reflected in the configuration file (APACHE$JAKARTA_CONFIG.DAT). In the future, you might want to have a development Tomcat server and a production Tomcat server on the same system, but with different configuration information for each server.