

---

# Installing and Getting Started with the Covalent Enterprise Ready Server

---



Release 2.4, January, 2004

---

Installing and Getting Started with the Covalent Enterprise Ready Server

---

---

# Installing and Getting Started with the Covalent Enterprise Ready Server

Release 2.4

Copyright © 2004 Covalent Technologies, Inc, 303 Second Street, Suite 375 South, San Francisco, CA 94107.  
All rights reserved.

This product and documentation are protected by copyright and distributed under licenses restricting their use, copying distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Covalent Technologies, Inc. and its licensors, if any. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

THIS PUBLICATION AND PRODUCT ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. COVALENT TECHNOLOGIES, INC. AND ITS LICENSORS DO NOT WARRANT THE RESULTS OBTAINED WITH THE USE OF THE PRODUCT OR THIS PUBLICATION NOR THAT THE PRODUCT OR THE OPERATION THEREOF WILL BE UNINTERRUPTED, ERROR-FREE OR VIRUS-FREE.

IN NO EVENT WILL COVALENT TECHNOLOGIES, INC. OR ITS LICENSORS BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OR FOR ANY CLAIM ATTRIBUTABLE TO ERRORS, OMISSIONS, OR OTHER INACCURACIES IN THIS PUBLICATION OR PRODUCT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY MADE OR ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED INTO NEW EDITIONS OF THE PUBLICATION. COVALENT TECHNOLOGIES, INC., MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Covalent and the Covalent Technologies, Inc. logo are trademarks of Covalent Technologies, Inc. in the U.S. and other countries and may not be used or reproduced without the prior written permission of Covalent Technologies, Inc. All other registered and unregistered trademarks in this document are the sole property of their respective owners.

Release 2.4, 19-01-04

# Contents



Introducing the Covalent Enterprise Ready Server . . . . .	1
Covalent Enterprise Ready Server Services . . . . .	1
Covalent Content Services . . . . .	1
Covalent Security Services . . . . .	1
Covalent Application Services . . . . .	1
Installation Requirements . . . . .	3
Platforms and Hardware Requirements . . . . .	5
Downloading and Unpacking the Covalent Enterprise Ready Server . . . . .	7
Downloading and Unpacking . . . . .	7
Installing and Getting Started with an Covalent Enterprise Ready Server . . . . .	9
Using Covalent Security Services . . . . .	12
Starting the Covalent Enterprise Ready Server . . . . .	12
Stopping the Covalent Enterprise Ready Server . . . . .	13
Restarting the Covalent Enterprise Ready Server . . . . .	13
Starting the Covalent Enterprise Ready Server with Covalent SSL . . . . .	14
Testing Covalent Enterprise Ready Server Modules . . . . .	15
Perl Module Examples . . . . .	15
Viewing the Examples and Learning More about the mod_perl Module . . . . .	15
Extra Examples . . . . .	17
Java Servlet API Examples . . . . .	17
JavaServer Pages Examples . . . . .	18
Viewing the Examples and Learning More about JSPs: . . . . .	18
On-line Documentation . . . . .	19
Security Issues with the Examples . . . . .	20
Troubleshooting . . . . .	21

# Contents

---

# Introducing the Covalent Enterprise Ready Server



---

The Covalent Enterprise Ready Server (ERS) is an instance of an Apache 2.0 Web server with Covalent extensions.

## Covalent Enterprise Ready Server Services

The Covalent Enterprise Ready Server provides the following services:

### Covalent Content Services

- Apache 2.0 HTTP Server
- Covalent Compatibility
- Covalent SNMP

### Covalent Security Services

- Covalent Authentication and Authorization
- Covalent SSL

### Covalent Application Services

- JavaServer Pages™/Servlets Support
- Perl Scripting Support

# Introducing the Covalent Enterprise Ready Server

---

## Updates to the Documentation

For the current version of this guide, refer to the on-line version at [www.covalent.net/support/enterprise\\_ready](http://www.covalent.net/support/enterprise_ready)

## About This Document

### Pathnames

This document uses UNIX pathnames throughout. Typically, the Covalent Enterprise Ready Server is installed in `/usr/local/covalent/ers` on UNIX systems and in `C:\Covalent\Ers` on Windows systems. These paths are the default installation directories for their respective operating systems. The pathname `/path/to/` as used in this documentation is a short-hand for the installation directory, regardless if you have used the default.

All paths are relative from the installation directory unless otherwise explicitly noted.

### Libraries

Library files on most UNIX servers (for example, Solaris, Linux, and others) have a `.so` extension. On HP-UX, the libraries have a `.sl` extension. On Windows, the libraries have a `.dll` extension. This document will use the standard UNIX extension, `.so`, throughout.

---

## Related Documentation

*Using the Covalent Enterprise Ready Server*

## Technical Assistance

Covalent offers a variety of support plans for its Apache-based products. From a business-hours Single Server plan to 24 x 7 coverage of server farms and complex applications, Covalent offers a plan that meets the needs for responsive, world-class support of Covalent Apache products. Covalent also offers an on-line Knowledge Base and FAQ database for licensed customers as well as registered evaluation customers.

Visit [www.covalent.net/support](http://www.covalent.net/support) for further information.

## Installation Requirements

For UNIX systems:

- The X Window System for the Graphical User Interface (GUI) version of the installer and other tools.
- The `gzip/gunzip` decompression utility.
- Superuser (root) access or knowledge of the root password to obtain superuser privileges with the `su` command.

### Notes

*If you do not have root permissions, refer to "Troubleshooting" on page 21.*

*On the HP-UX platform, you must define the user and group `www` before you install this product.*

*On Solaris 8, you must install all patches recommended for running `j2se1.4` as specified at: <http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access>*

## Introducing the Covalent Enterprise Ready Server

---

*On Linux systems, libdb.so.3 must be available. Many current releases of Linux ship with libdb.so.2. You must replace this library before the Apache Webserver will successfully start. Caldera Linux has libdb-3.3.so, which is a symbolic link to libdb-3.3.so.3.3.11. You must create a new symbolic link to libdb-3.3.so.3.3.11 named libdb.so.3.*

*Due to licensing constraints, ERS does not include a JSDK for HP-UX. In order for the jsp examples to function you must obtain the JDK separately for HP-UX and configure ERS to see it.*

For Windows systems:

- Administrator access or a access to a user account with Administrator privileges.
- A decompression utility such as PKZIP or WinZIP.

---

## Platforms and Hardware Requirements

All machines on which you wish to install ERS must have static IP addresses. Enterprise Ready Servers bind to specific IP/port combinations.

### Currently Supported Platforms

- Sparc Solaris 2.6,7,8
- Redhat 7.1, 7.2
- HPUX 11
- Windows 2000

### System Requirements

- CPU: 800MHz
- Memory: 256MB minimum, 512MB recommended
- Hard Disk Space: 500MB

## Introducing the Covalent Enterprise Ready Server

---

# Downloading and Unpacking the Covalent Enterprise Ready Server



## Downloading and Unpacking

- 1 If you do not have the Covalent Enterprise Ready Server Release 2.4 distribution, purchase the full release at:

[www.covalent.net/products/enterprise\\_ready](http://www.covalent.net/products/enterprise_ready)

- 2 Download the distribution using the instructions you received when you completed your purchase.
- 3 The distribution will be a `.tar.gz` file for UNIX systems and a `.zip` file for Windows systems. You need to unpack it in a temporary directory.
  - On a UNIX system, execute the following command(s):

```
gunzip -c ers-2.4-platform.tar.gz | tar -xvf -
```

You need to substitute your operating system for *platform* in the above command. For example, `ers-2.4-sparc-sun-solaris-8.x.tar.gz`. Note that spaces and capitalization are significant.

- On a Windows system, double-click the `ers-2.4-platform.zip` file from the Windows Explorer.

To install the Covalent Enterprise Ready Server, proceed to "[Installing and Getting Started with an Covalent Enterprise Ready Server](#)" on page 9

## Downloading and Unpacking the Covalent Enterprise Ready Server

---

# Installing and Getting Started with an Covalent Enterprise Ready Server



---

To install ERS, change to the directory created by unpacking the Covalent Enterprise Ready Server Release 2.4 distribution (described in "[Downloading and Unpacking the Covalent Enterprise Ready Server](#)" on page 7).

- 1 Execute the `setup` command:
  - On Unix systems, the command is `./setup`
  - On Windows systems, double click on `setup.exe` or type `setup.exe` on the command line.
- 2 Select the installation for the **Covalent Enterprise Ready Server**.
- 3 Click **Next** to continue.
- 4 Specify the directory into which you want the ERS to be installed.
  - On UNIX systems, the default directory is `/usr/local/covalent/ers-2.4`
  - On Windows systems, the default directory is `C:\covalent\ers-2.4`
- 5 Click **Next** to continue.
- 6 Choose the type of installation:
  - **Typical**

This installs a default configuration. It is recommended for most users.
  - **Custom**

This allows you to select the modules you want to install. It should be selected by advanced users only. Apache 2.0 is installed by default.

## Installing and Getting Started with an Covalent Enterprise Ready Server

---

7 For a **Custom** installation, select the modules you want installed and activated along with the Apache 2.0 HTTP server. The available modules are:

- Apache (installed automatically)
- Covalent Enterprise Ready Server Configuration
- Covalent Authentication and Authorization
- Covalent SNMP
- Covalent SSL
- Tomcat (JavaServer Pages/Servlets)
- Java Software Development Kit
- Perl Scripting (`mod_perl`)

See "[The Covalent Enterprise Ready Server \(ERS\) is an instance of an Apache 2.0 Web server with Covalent extensions.](#)" on page 1 for a list of the available modules. For detailed descriptions, refer to *Using the Covalent Enterprise Ready Server*.

8 You are prompted for the configuration information:

- Server Name

If the installer cannot determine your system host name, the default is `localhost`. Otherwise it is the host name.

- Port

This is the port the Covalent Enterprise Ready Server will listen on for requests. The default is port 80. If you are installing as a non-root user, the default is 8080.

- Server Administrator Email

The Covalent Enterprise Ready Server will use this address for critical messages.

9 On UNIX systems, you are prompted to select the Multi-Processing Modules (MPM) you want as the default:

- 
- Prefork MPM

This model implements a non-threaded, pre-forking server. It handles requests in a manner very similar to the default behaviour of Apache 1.3. This server is very robust.

- Worker MPM

This model implements a hybrid multi-process multi-threaded server for systems that support POSIX threads. Each process has a fixed number of threads. When a request is received, it is passed to a worker thread for processing. The server adjusts to changes in its load by increasing or decreasing the number of processes. This server scales very easily but emphasizes robustness.

If you are running Windows, this selection is not available because there is only one MPM available for Windows systems.

- 10 If you selected a **Typical** installation or you selected to install the Covalent SNMP module, you are prompted to select the versions you want to support. You can choose from:

- All available SNMP versions
- Only SNMP version 3 (SNMPv3)
- Only SNMP version 1 (SNMPv1) and 2c (SNMPv2c)

- 11 If you selected "All available SNMP versions" or "Only SNMP version 1 and 2c" in the previous step, you are prompted to select remote or local access to the Covalent SNMP module.

If you select remote access, you are prompted to provide the network address and subnet mask to enable remote access to the Covalent SNMP module. The default network address is 10.0.0.0 and the default subnet mask is 255.255.255.0.

If you select local access, you can only access the Covalent SNMP module from the localhost interface.

- 12 You are prompted for a username and password.

This password is used as the default password for Covalent Authentication and Authorization to provide SNMP v3 authorization, and to protect the dynamic examples for the Tomcat (JavaServer Pages/Servlets) and Perl Scripting modules.

## Installing and Getting Started with an Covalent Enterprise Ready Server

---

The default username is `covalent`. The password you supply must be at least eight characters long. It should also contain some or all of the following: mixed case, numbers, and punctuation characters. Avoid dictionary words or words that could be easily guessed.

**Be sure to remember this password!**

- 13 The installer asks you to confirm your selections and then proceeds to copy the files to the installation directory.

Upon completion, the installer prompts you to start the Covalent Enterprise Ready Server. If you elect not to start it from the installer, follow the instructions in ["Starting the Covalent Enterprise Ready Server" on page 36](#).

## Using Covalent Security Services

Before you can use any of the Covalent Security Services, you must configure them. To learn more about configuring Covalent Authentication and Authorization, refer to the section entitled *Security Services* in *Using the Covalent Enterprise Ready Server*. Appendix B of *Using the Covalent Enterprise Ready Server* describes the SSL configuration directives in more detail.

If you installed Covalent SSL, you must first generate a private key and temporary server certificate before you can run the server with SSL. See ["Generating a Server Certificate and Private Key" on page 23](#) for further instructions.

## Starting the Covalent Enterprise Ready Server

On UNIX servers, start the Covalent Enterprise Ready Server with the following commands:

```
/path/to/servers/${ServerName}/bin/tomcat_startup.sh start
/path/to/servers/${ServerName}/bin/apache_startup.sh start
```

### Note

*To accommodate custom configurations, the startup commands for UNIX systems are located in the `/path/to/servers/${ServerName}` directory.*

---

On Windows systems, start the Covalent Enterprise Ready Server from the Services panel. It is controlled by the `Covalent$(ServerName)ApacheERS2.4` service. By default, it is set to **Automatic** start-up.

The Jakarta Tomcat container is controlled by the `Covalent$(ServerName)TomcatERS2.4` service. It is also set to **Automatic** start-up.

## Stopping the Covalent Enterprise Ready Server

On UNIX systems, stop the Covalent Enterprise Ready Server with the following commands:

```
/path/to/servers/$(ServerName)/bin/apache_startup.sh stop  
/path/to/servers/$(ServerName)/bin/tomcat_startup.sh stop
```

On Windows systems, stop the Covalent Enterprise Ready Server from the Services panel. Select the `Covalent$(ServerName)ApacheERS2.4` service and click **Stop**. Then select the `Covalent$(ServerName)TomcatERS2.4` service and click **Stop**.

## Restarting the Covalent Enterprise Ready Server

If you must restart a running Enterprise Ready Server, you can use the **Restart** command to do so.

On UNIX servers, if you restart when the SNMP module is loaded, a hard restart is performed. If you stop the SNMP module and then restart, the system waits for all requests to be serviced, and then stops the child processes, reloads the configuration files, and restarts the processes.

Restart the Covalent Enterprise Ready Server with the following command:

```
/path/to/servers/$(ServerName)/bin/apache_startup.sh [restart | graceful]
```

On Windows systems, restart the Covalent Enterprise Ready Server from the Services panel. Select the `Covalent$(ServerName)ApacheERS2.4` service and click **Restart**. The system waits for all requests to be serviced, and then stops the child processes, reloads the configuration files, and restarts the processes.

### Starting the Covalent Enterprise Ready Server with Covalent SSL

The Covalent Enterprise Ready Server `httpsd.conf` file is configured correctly by default to use the server certificate and private key you create using the instructions in Using the Enterprise Ready Server.

- 1 If you installed Covalent SSL on a UNIX system, start the Covalent Enterprise Ready Server with the following command:

```
/path/to/servers/${ServerName}/bin/apache_startup.sh startssl
```

On Windows systems, the Covalent Enterprise Ready Server with Covalent SSL is started from the Services panel. You must first change the Windows Service options from a Windows command shell to enable Covalent SSL. For example:

```
\path\to\apache\bin\httpsd -k config -d \path\to\servers\${ServerName} -D SSL  
-n "Covalent ${ServerName} ApacheERS2.4"
```

#### Note

*Once you have made this change, the Covalent Enterprise Ready Server will always start with Covalent SSL. You must run the above command again, without the `-D SSL` option to re-enable the standard HTTP server.*

During server start-up on UNIX systems, you will be prompted to enter the pass phrase for the (temporary) self-signed server certificate. By default, the server certificate is not assigned a pass phrase on Windows systems.

- 2 To test your certificate, start a browser and go to `https://myserver.mydomain/`

The browser warns you that it does not recognize the signature on your certificate.

- 3 Choose to accept the certificate.

The browser displays the certificate.

# Testing Covalent Enterprise Ready Server Modules



---

Once you have installed an Enterprise Ready Server, you can use the example pages included with the product to test any modules you have chosen to install. If you installed `mod_perl` or either Java Servlet container, use this section as a guide to the examples.

## Note

*The first time you access an example page for `mod_perl`, Java Servlets, or JavaServer Pages, you will be prompted for a username and password. Enter the username and password you defined during the installation process. If your browser caches this information, you should not be prompted again as you try the different examples during a browser session.*

## Perl Module Examples

The `mod_perl` examples included with Covalent Enterprise Ready Server show a variety of ways you can use `mod_perl` with your server.

### Viewing the Examples and Learning More about the `mod_perl` Module

- 1 Start a browser and go to `http://myserver.mydomain/`

The Covalent Enterprise Ready Server start page is displayed in your browser.

- 2 Under the Examples heading, select **Mod Perl**.

The Mod Perl Examples page is displayed.

The Covalent Enterprise Ready Server contains three `mod_perl` examples:

## Testing Covalent Enterprise Ready Server Modules

---

- "Mod Perl as a CGI replacement"
- "Mod Perl as Apache Perl Module development environment"
- "Extra examples"

### Using Mod Perl as a CGI Replacement

The examples for using `mod_perl` as a CGI replacement demonstrate three ways to convert CGI processes to `mod_perl`:

- CGI Compatible Mode

Example of a CGI script converted to run as a `mod_perl` script.

- Apache API Mode

Example of an embedded `mod_perl` running within the Apache server.

- CGI.pm Mode

Example of a `mod_perl` module script running as a Web application. For more information about CGI.pm, see [stein.cshl.org/WWW/software/CGI/cgi\\_docs.html](http://stein.cshl.org/WWW/software/CGI/cgi_docs.html).

The output displays the same text, however, each function executes differently. All three examples depend on the `Apache::Registry` module being loaded. Select the **Source Code** links to see the differences in each `mod_perl` example.

### Using Mod Perl as an Apache/Perl Module Development Environment

These examples demonstrate `mod_perl` running a Perl module to:

- Perform a "Hello World" example.
- Execute a Perl script to perform authentication.

The Echo example displays the "Hello World" page, while the authentication example requests a username (`covalent`) and password (`handy`) before displaying a Restricted Area page.

---

## Extra Examples

The final `mod_perl` example illustrates how to complete a form and use the information from the form on another page.

After you select **Run Example**, complete the form and select **Submit** to see how the information you provided is used. Leave some or all of the blanks empty to see the resulting message.

## Java Servlet API Examples

The Java Servlet API examples included with Covalent Enterprise Ready Server demonstrate a variety of ways you can implement Java Servlets using the Jakarta Tomcat container.

To view the examples and learn more about Java Servlets:

- 1 Start a browser and go to `http://myserver.mydomain/`

The Covalent Enterprise Ready Server start page is displayed in your browser.

- 2 Under the Examples heading, select **Java Servlet API**.

The Servlet Examples with Code Web page displays.

The Servlet Examples with Code page contains six Java Servlet functions. Select **Execute** to run an example and select **Source** to view its code.

The table below describes each Servlet example:

---

Hello World	This example demonstrates a Hello World example.
Request Info	This example demonstrates using Java Servlets to encapsulate information received from the client with the <code>HttpServletRequest</code> method.
Request Headers	This example demonstrates using Java Servlets to handle HTTP requests.

---

## Testing Covalent Enterprise Ready Server Modules

---

---

Request Parameters	This example demonstrates how Java Servlets can access parameters passed into a Java Servlet during a request, then format the information and send it back to the client as HTML.
Cookies	This example demonstrates using a Java Servlet for HTTP and HTTPS session management.
Sessions	This example demonstrates using Java Servlets to get information from a user's session.

---

## JavaServer Pages Examples

The JavaServer Pages examples included with Covalent Enterprise Ready Server demonstrate a variety of ways you can implement JSPs using the Jakarta Tomcat container.

### Viewing the Examples and Learning More about JSPs:

- 1 Start a browser and go to `http://myserver.mydomain/`

The Covalent Enterprise Ready Server start page is displayed in your browser.

- 2 Under the Examples heading, select **Java Server Pages (JSP)**.

The JSP Examples page is displayed.

The JSP Examples page contains 13 JSP functions. Select **Execute** to run an example and select **Source** to view its code.

The table below describes each JavaServer Page example:

---

Numberguess	This example demonstrates interacting with a Web page where you submit information and the Web page outputs information based on what you submitted.
Date	This example demonstrates how JavaServer Pages can be used to display the date.

---

---

Snoop	This example demonstrates how to obtain Request Information using JavaServer Pages.
ErrorPage	This example demonstrates using JavaServer Pages for error processing.
Carts	This example demonstrates using JavaServer Pages to create a shopping cart on your Web site.
Checkbox	This example demonstrates selecting items from a list and submitting the list.
Color	This example demonstrates using JavaServer Pages and JavaBeans.
Calendar	This example demonstrates a basic personal planner function that allows you to add daily appointments to your calendar via the Web.
Include	This example demonstrates including the current time on a Web page.
Forward	This example demonstrates showing the virtual memory.
JSP-Servlet-JSP	This example demonstrates using a Java Servlet to call another servlet.
Simple Custom Tag	This example demonstrates using custom tags with JavaServer Pages to create dynamic output.
Sessions	This example demonstrates using JavaServer Pages to get information from a user's session.

---

## On-line Documentation

The `mod_perl` and Jakarta-Tomcat modules are developed as open-source projects each with their own documentation. For your convenience, Covalent has included links to each module's documentation on the Covalent Enterprise Ready Server page.

To view the on-line documentation:

- 1 Start a browser and go to `http://myserver.mydomain/`

The Covalent Enterprise Ready Server start page is displayed in your browser.

## Testing Covalent Enterprise Ready Server Modules

---

- 2 Under the On-line Documentation heading, select the links to view the project documentation for each module listed:
  - **Mod Perl** - Select this link to go to the [Apache/Perl Integration Project](#) site and check the [Books and Documentation](#) links to find a list of available documentation about `mod_perl`. Also see the [mod\\_perl Guide](#).
  - **Jakarta Tomcat** - Select this link to go to the [Jakarta Project](#) site. Navigate to the [Tomcat Project](#). From there, select the [Tomcat 4.0](#) link under Documentation to view the project documentation.
  - **Covalent SSL Documentation** - See [www.covalent.net/support/ssl](http://www.covalent.net/support/ssl).

## Security Issues with the Examples

To keep out unauthorized users, access to the Covalent Enterprise Ready Server example pages is secured with a username and password. You should take the following steps after you complete testing:

- Read the documentation for the modules you installed for details about the security risks of running each module.
- Determine whether to retain or remove password protection from the examples, based upon your security concerns and risks.
- If you decide to completely disable the examples, in the `httpsd.conf` file type # in front of the `Alias` directive associated with the `mod_perl` and Java examples to comment out the directive and prevent browser access. For example:

```
# Alias /examples/modperl /path/to/modperl/examples/modperl
```

### Note

*Be sure to restart your server after you modify the `httpsd.conf` file.*

# Troubleshooting

---



If you do not have root permission on UNIX systems:

- You can still install Covalent Enterprise Ready Server, however the automatic process cannot complete all of the installation steps. The installer displays warnings to help you determine which files you need to edit by hand.
- The installer will set the default location for the Covalent Enterprise Ready Server to `covalent/ers` under your home directory.
- Only the superuser, root, is allowed to bind to ports less than 1024 which mean your server will not be allowed to bind to ports 80 and 443. Therefore, the installer sets the default port for HTTP to 8080 and for HTTPS to 8443. This enables you to test your server without editing the `httpsd.conf` file. Start up a browser and test your server with `http://myserver.mydomain:8080/` and `https://myserver.mydomain:8443/`.
- Be sure to edit `httpsd.conf` to use the standard ports for HTTP (port 80) and HTTPS (port 443) before your secure site goes live.

## Note

*Non-root installations are not recommended for SSL-enabled servers. A non-root installation is less secure because it allows anybody with root permissions access to your keys and certificates.*

If you have problems starting your server:

- Verify that the `ServerName` directive in `httpsd.conf` corresponds to the name of your server.
- Verify that the server is not already running.

## Troubleshooting

---

- On UNIX systems, execute:

```
ps -ax | grep http
```

on BSD compatible systems, or:

```
ps -elf | grep http
```

on System V Release 4 compatible systems.

If this results in a list of processes, stop the running server before starting your new server.

- On Windows systems, start Services and check for the Covalent Apache process.
- Check the `error_log` file in the `/path/to/servers/${ServerName}/logs` directory. Also check the `covalent_error_log` file as well as the `covalent_ssl_log` if your server is running Covalent SSL.
- Check the syntax of your configuration file:

```
/path/to/apache/bin/httpsdc -t
```

The server should respond with `syntax ok`.

### Changing the Java JDK used by the Covalent Enterprise Ready Server

- The Covalent Enterprise Ready Server is supplied with the Java™ 2 SDK, Standard Edition 1.4. It is installed in the directory `/path/to/jsdk`. If you want to use another version of the Java2 SDK, you must edit the Covalent Enterprise Ready Server start-up file `/path/to/servers/${ServerName}/bin/tomcat_startup.sh`.
  - 1 Scroll down to the line that defines `JAVA_HOME`.
  - 2 Replace the value in the `JAVA_HOME` definition with the location of the Java SDK you want to use.

#### **CAUTION!**

**Any Java SDK you define is not guaranteed to work with the 4.1 Tomcat version as supplied with the Covalent Enterprise Ready Server.**

---

## Changing the MPM used by the Covalent Enterprise Ready Server

- On UNIX systems, the Covalent Enterprise Ready Server is installed with two Multi-Processing Modules (MPM), Prefork MPM and Worker MPM, which are installed in `/path/to/apache/bin/` as `httpsd.prefork` and `httpsd.worker` respectively. There is a start-up script for each server, `/path/to/servers/$(ServerName)/bin/apache_startup.sh`. There is also a file, `startup.properties`, defining the specific properties for the server in the directory `/path/to/servers/$(ServerName)/conf/`. This file contains a variable, `default_mpm`, which specifies the MPM the server is to use.

The following is an example of the `startup.properties` file:

```
# Covalent server properties file
#
# default_mpm specifies which MPM the server should run under. The
# possible values are:
#
# prefork - A non-threaded multi-process server
#
# worker - A hybrid multi-process multi-threaded server
default_mpm="worker"
#
# Flags to be passed when starting Apache.
#
default_flags=""
```

The `default_flags` variable is a dummy that can be used to pass extra arguments to the `httpd` process, for example `-DONE_PROCESS`.

## Setting the Temporary Directory for the Installer on UNIX systems:

- On UNIX systems, the installer uses the directory `/var/tmp` to store its temporary files regardless the setting of the `$TMPDIR` environment variable (if set). If your system has limited space available in this directory, the installation may fail. Use the `-is:tmpdir` command line option to change the temporary directory setting. For example:

```
setup -is:tmpdir /path/to/tmp
```

## Troubleshooting

---

If you are using Solaris and encounter problems running the Covalent Apache binaries:

- If you plan to support Java Servlets or JavaServer Pages, verify that you have installed all recommended operating system patches for your version of Solaris. See [java.sun.com/j2se/1.4.2/relnotes.html](http://java.sun.com/j2se/1.4.2/relnotes.html) for information about Solaris related patches.
- The default shared memory configuration for Solaris is probably inadequate. Use `sysdef` to check your allocation. If "max shared memory segment size (SHMMAX)" is only 1MB or so, you'll need to adjust it. Add the following to `/etc/system` then reboot:

```
set shmsys:shminfo_shmmax = 0x2000000
set shmsys:shminfo_shmmni = 0x1000
set shmsys:shminfo_shmseg = 0x100
set shmsys:shminfo_shmmin = 1
set semsys:seminfo_semmni = 0x200
set semsys:seminfo_semmns = 0x200
set semsys:seminfo_semmns1 = 0x20
```

If you are using HP-UX and encounter problems running the Covalent binaries:

- HP-UX does not allow the user `nobody` access to system shared memory. Therefore, the `httpd` daemon and associated binaries must be owned by the user `www`. By default, the `www` user and group are not defined on the HP-UX platform.
- HP-UX systems are by default configured with extremely low shared memory allocations. Use SAM (`/usr/sbin/sam`) to examine and adjust your allocations. Select "Kernel Configuration" and change the following parameters:
  - `semmni` from 64 to 256
  - `semmns` from 128 to 512
  - `max_thread_proc` from 64 to 1024
  - `maxfiles` from 60 to 256
  - `nkthread` from 499 to 3635
  - `nproc` from 276 to 2068
  - `ncallout` from 292 to 2084

You can also change `maxusers` from 32 to 256. This parameter adjusts a number of other values in the HP-UX kernel.

---

After you have made your changes, use the "Process New Kernel" command to make the changes permanent.

- Due to licensing constraints, ERS does not include a JSDK for HP-UX. In order for the jsp examples to function you must obtain the JDK separately for HP-UX and configure ERS to see it.

If you are using Red Hat Linux 7.1 and encounter problems running the Covalent Apache binaries:

- If you plan to support Java Servlets or JavaServer Pages, you need to update the GNU C library (glibc) to version 2.4. To determine which version of glibc is installed on your system, use the following command:

```
rpm -q glibc
```

You can download a more recent version of glibc from the [Red Hat](#) website. You can also use [RPMFind](#) to locate an appropriate version.

