Sun Web Stack Getting Started Guide

Sun Microsystems Inc.

Part No: 820–6720
2008

December 2008

4150 Network Circle

Santa Clara, CA 95054

U.S.A.

081218@21808

Contents

CHAPTER 1

Getting Started with Sun Web Stack

This document provides an overview of the Sun Web Stack software and how it can be configured in Solaris 10 OS and Red Hat Enterprise Linux (RHEL).

Sun Web Stack Software

The Sun Web Stack software provides a collection of open source servers and interpreters that are bundled and made available for various platforms including Solaris 10 OS, OpenSolaris OS, and Red Hat Enterprise Linux (RHEL).

After downloading the Sun Web Stack software, follow the instructions provided in the Sun Web Stack Installation and Configuration Guide for information on installing and configuring the stack on specific platform.

The Sun Web Stack software is supported on the following platforms:

1. **Solaris 10 OS** (Update 5 or higher).
2. **OpenSolaris 2008.11 OS**.
3. **Linux** (Red Hat Enterprise Linux (RHEL) 5 Update 2 or higher (32-bit and 64-bit kernels)). Support is available only for x86 platforms.

Note – This document covers only the Solaris 10 OS and Red Hat Enterprise Linux platforms. For using Sun Web Stack software in OpenSolaris 2008.11 OS, see the OpenSolaris 2008.11 Web Stack Getting Started Guide.

This document provides you with information on how you can use Sun Web Stack software on the supported platforms after you have successfully installed and configured the software.

The Sun Web Stack software includes the following:

1. **Server Software** - Apache Web Server, Apache Tomcat, Squid Proxy Server, lighttpd.

2. **DB Software** - MySQL Server.
3. **Programming Environment** - PHP, Perl, Ruby, and Python.
4. **Caching Software** - Memcached.

Ruby and Python are not provided as part of the Sun Web Stack software for RHEL. You need to use the interpreters provided by the Red Hat platform.

The following section shows you how you can download and configure the Sun Web Stack software on your platform.

Getting the Sun Web Stack Software

The Sun Web Stack binary packages are available for all the supported platforms. For a list of supported platforms, refer to the previous section.

For more information on installing and configuring the Sun Web Stack software for your platform, see theses documents:

- Installing Web Stack
- Post Installation Configuration
- Advanced Configuration
- Upgrading Sun Web Stack

Packages available through the Sun Web Stack Software

The following table shows the list of all packages available through the Sun Web Stack software in Solaris 10 OS:

Component Name	Package Name
Apache HTTP Server 2.2.9	`sun-apache22`
mod_dtrace 0.3.1	`sun-apache22-dtrace`
mod_fcgid 2.2	`sun-apache22-fcgid`
mod_jk 1.2.25	`sun-apache22-jk`
mod_perl 2.0.2	`sun-apache22-perl`
-	-
lighttpd 1.4.19	`sun-lighttpd14`

Component Name	Package Name
memcached 1.2.5	sun-memcached
-	-
MySQL 5.0.67	sun-mysql50
Regression tests for MySQL	sun-mysql50test
-	-
PHP 5.2.6	sun-php52
APC Extension	sun-php52-apc
IDN Extension	sun-php52-idn
Memcache Extension	sun-php52-memcache
Suhosin Extension	sun-php52-suhosin
TCPWrap Extension	sun-php52-tcpwrap
XDebug Extension	sun-php52-xdebug
DTrace Extension	sun-php52-dtrace
-	-
Python 2.5.2	sun-python25
Ruby 1.8.6	sun-ruby18
Squid 2.6.STABLE17	sun-squid
Apache Tomcat 5.5.27	sun-tomcat5
Support libraries and registration tool	sun-wsbase

Note – Always use the Sun Web Stack installer to install individual packages or the whole stack.

See the Sun Web Stack Installation and Configuration Guide.

The following table shows the list of all packages available through the Sun Web Stack software for RHEL:

Apache HTTP Server 2.2.9	sun-apache22

mod_fcgid 2.2	`sun-apache22-fcgid`
mod_jk 1.2.25	`sun-apache22-jk`
mod_perl	`sun-apache22-perl`
-	-
lighttpd 1.4.19	`sun-lighttpd14`
memcached 1.2.5	`sun-memcached`
-	-
MySQL 5.0.67	`sun-mysql50`
Regression tests for MySQL	`sun-mysql50test`
-	-
PHP 5.2.6	`sun-php52`
APC Extension	`sun-php52-apc`
IDN Extension	`sun-php52-idn`
Memcache Extension	`sun-php52-memcache`
Suhosin Extension	`sun-php52-suhosin`
TCPWrap Extension	`sun-php52-tcpwrap`
XDebug Extension	`sun-php52-xdebug`
-	-
Squid 2.6.STABLE17	`sun-squid`
Apache Tomcat 5.5.27	`sun-tomcat5`
Support libraries and registration tool	`sun-wsbase`

Note – Always use the Sun Web Stack installer to install individual packages or the whole stack.

The following list provides you with information on what is **not supported** by the Sun Web Stack software on Red Hat platform:

1. Ruby support (use the OS-provided Ruby interpreter).
2. Python support (use the OS-provided Python interpreter).
3. `mod_dtrace` plug-in module for Apache (Linux doesn't support DTrace).
4. Dtrace extension for PHP (Linux doesn't support DTrace).

See the Sun Web Stack Installation and Configuration Guide.

Installing the Entire Sun Web Stack Bundle

See the Sun Web Stack Installation and Configuration Guide.

Setting Up Your Server Environment

If you have installed the Sun Web Stack packages by using the amp installation option, the Sun Web Stack software will be available under /opt/sun/webstack for RHEL and /opt/webstack for Solaris 10 OS.

For more information on the configuration file layout, see the Installation and Configuration Guide

The Sun Web Stack software comes with Apache 2.2.9 Web Server fully configured and optimized for PHP deployment. The NetBeans IDE 6.0 lets you add the Apache Web Server in the servers list so the applications that you build including PHP based web applications can be deployed to the Apache Web Server. To set up the Apache Web Server in the NetBeans IDE, refer to the later sections in this document.

To start the Apache Web Server on Solaris 10 OS, execute the following command:

```
/usr/sbin/svcadm enable sun-apache22
```

You need to have the necessary permission to execute the previous command.

To start the Apache Web Server on Red Hat Enterprise Linux, execute the following command:

```
/sbin/service sun-apache22 start
```

or

```
/etc/init.d/sun-apache22 start
```

Where is the Apache Web Server Installed?

The following table shows the locations where you can find the Sun Web Stack files:

Platform	Install Location
Solaris 10 OS	/opt/webstack/apache2/2.2/
RedHat Enterprise Linux	/opt/sun/webstack/apache2/2.2/

Setting Up the MySQL DB

The MySQL server is the most widely used relational database server, and the MySQL package is available through the Sun Web Stack software. When you install the amp configuration, you are also installing the MySQL database. You can install the complete amp configuration as shown below:

```
$ ./install amp
```

or just install the MySQL DB:

```
$ ./install mysql
```

You still need to configure the MySQL DB in order for it to function. See the Installation and Configuration Guide for more information.

Where is the MySQL DB Installed?

The following table shows the locations where you can find the Sun Web Stack files:

Platform	Install Location
Solaris 10 OS	/opt/webstack/mysql/5.0
RedHat Linux Enterprise Linux	/opt/sun/webstack/mysql/5.0

To start the MySQL Server on Solaris 10, execute the following command:

```
/usr/sbin/svcadm enable sun-mysql50
```

You need to have the necessary permission to execute the previous command.

To start the Apache Web Server on Red Hat Enterprise Linux, execute the following command:

```
/sbin/service sun-mysql50 start
```

or

```
/etc/init.d/sun-mysql50 start
```

More MySQL Server Information

The default password for the MySQL Server root user is a blank string (no password). After starting the MySQL Server using the command provided in the previous section, you can connect to the MySQL Server using the following command:

For **Solaris 10** OS:

```
/opt/webstack/mysql/5.0/bin/mysql -u root
```

For **RHEL** OS:

```
/opt/sun/webstack/mysql/5.0/bin/mysql -u root
```

For changing the default root password, perform the following step:

For **Solaris 10** OS:

```
/opt/webstack/mysql/5.0/bin/mysqladmin -u root password {{<new-pass>}}
```

For **RHEL** OS:

```
/opt/sun/webstack/mysql/5.0/bin/mysqladmin -u root password {{<new-pass>}}
```

Caution – Change the default MySQL Server root password before deploying it in a production environment.

Automatic Startup of Servers

In Solaris 10 OS, the Apache Web Server automatically starts when the machine restarts if you enable the sun-apache22 service.

In RHEL, startup script sun-apache22 is available under /etc/init.d. To automatically start the service when the machine starts, use the chkconfig mechanism.

To start the Apache Web Server in RHEL when the machine starts, execute the following command:

```
$ /sbin/chkconfig sun-apache22 on
```

To start the MySQL DB in RHEL when the machine starts, execute the following command:

```
$ /sbin/chkconfig sun-mysql50 on
```

To start the MySQL DB in RHEL when the machine starts, execute the following command:

```
$ /sbin/chkconfig sun-mysql50 on
```

Note – chkconfig command only ensures that the startboot feature is enabled in RHEL. However, to start/stop the Apache Web Server or MySQL Server, you must use the service command in RHEL.

For instance,

```
$ sudo service sun-apache22 start
```

To start the Tomcat server in RHEL when the machine starts add the line /etc/init.d/sun-tomcat5 start to the end of the /etc/rc.d/rc.local file. For starting the Squid Proxy Server, add the line /etc/init.d/sun-squid start to the end of the /etc/rc.d/rc.local file.

Note – The default installation of the MySQL Server creates a file my.cnf under /etc/opt/webstack/mysql/5.0 (RHEL) or /etc/opt/sun/webstack/mysql/5.0 (Solaris 10 OS). This default configuration is only for very small deployments. There are other sample configurations under mysql/5.0/my-*.cnf that you can use for medium to large deployments. Move one of these files to my.cnf before starting the MySQL service

Setting Up a Secure Website

You were introduced to the basic Apache server configuration for setting up a web site. Now you will set up a secure web server to process web requests over https. mod_ssl, included with the Apache web server provides support for https.

To configure the SSL for your server, perform the following tasks:

1. Generate a Certificate
2. Modify the Apache configuration files

Generating a Certificate

For the purpose of this example, we will generate a (self-signed) test certificate. This certificate will not be accepted by a browser because it is not signed by a certificate authority (CA), which your browser is setup to trust.

To configure a real public secure server, you can pay a company such as Verisign to sign your certificate.

Note – You can only run one secure web server on a machine (if IP alias is not configured) because the certificates are server wide, and must contain the name of the site they represent. But you can let the server work on both secure and non-secure connections.

For **Solaris 10** OS, from a shell, run:

```
$ /usr/sfw/bin/openssl req -new -x509 -nodes -out server.crt -keyout server.key
```

For **RHEL**, from a shell, run:

```
$ /usr/bin/openssl req -new -x509 -nodes -out server.crt -keyout server.key
```

You will be asked some questions to fill the certificate's attributes. Read this document for more information.

The `openssl` utility is available in Solaris 10 OS and RHEL by default.

The previous command generates 2 files:

1. **server.crt** - SSL Certificate File
2. **server.key** - SSL Certificate Key File

Modifying the `ssl.conf` in Solaris 10 OS

In Solaris 10 OS, there is a sample `ssl.conf` file available under `/etc/opt/webstack/apache2/2.2/samples-conf.d`.

Move this file to `/etc/opt/webstack/apache2/2.2/conf.d`:

```
$ sudo cp /etc/opt/webstack/apache2/2.2/samples-conf.d/ssl.conf
/etc/opt/webstack/apache2/2.2/conf.d/ssl.conf
```

Then modify the `/etc/opt/webstack/apache2/2.2/conf.d/ssl.conf` file in order to change the values of these parameters:

```
...
SSLCertificateFile "<path to server.crt>"
SSLCertificateKeyFile "<path to server.key>"
...
```

Configuration files are read from `/etc/opt/webstack/apache2/2.2/conf.d` automatically. Thus in `/etc/opt/webstack/apache2/2.2/httpd.conf`, you will see the following line:

```
...
Include /etc/opt/webstack/apache2/2.2/conf.d/*.conf
...
```

Restart the Apache Web Server in order for the changes to take effect.

```
/usr/sbin/svcadm restart sun-apache22
```

Modifying the `ssl.conf` in RHEL

In RHEL, there is a sample `ssl.conf` file available under `/etc/opt/sun/webstack/apache2/2.2/samples-conf.d`.

Move this file to `/etc/opt/sun/webstack/apache2/2.2/conf.d`:

```
$ sudo cp /etc/opt/sun/webstack/apache2/2.2/samples-conf.d/ssl.conf
/etc/opt/sun/webstack/apache2/2.2/conf.d/ssl.conf
```

Then modify the `/etc/opt/sun/webstack/apache2/2.2/conf.d/ssl.conf` file in order to change the values of these parameters:

```
...
SSLCertificateFile "<path to server.crt>"
SSLCertificateKeyFile "<path to server.key>"
...
```

Configuration files are read from `/etc/opt/sun/webstack/apache2/2.2/conf.d` automatically. Thus in `/etc/opt/sun/webstack/apache2/2.2/httpd.conf`, you will see the following line:

```
...
Include /etc/opt/sun/webstack/apache2/2.2/conf.d/*.conf
...
```

Restart the Apache Web Server in order for the changes to take effect.

```
sudo /etc/init.d/sun-apache22 restart
```

From a browser window, invoke https://localhost. Some browsers complain of invalid security certificates. Add an exception and continue to test the secure connection.

- For Solaris 10 OS, the Apache configuration files will be available under `/etc/opt/webstack/apache2/2.2/conf.d`
- For Red Hat Enterprise Linux, the files will be available under `/etc/opt/sun/webstack/apache2/2.2/conf.d`
- OpenSSL certificates HOWTO
- mod_ssl module reference

Setting Up a Proxy Server

The Squid Proxy Server is both a proxy server and a web cache daemon available through the Sun Web Stack software. The Squid Proxy Server is primarily used to speed up a web server by caching repeated requests.

The Squid Proxy Server can be used in 2 ways:

1. The Squid Proxy Server can act in **Normal Mode** where it can cache the contents of an unlimited number of web servers for a limited number of clients.
2. The Squid Proxy Server can act in **Reverse Proxy Mode** (web server acceleration) where the cache serves an unlimited number of clients for a limited number of servers.

For instance, if your site is becoming slower because of heavy load there are several things you can do without buying additional hardware:

1. Configure your application to use `memcached`.
2. Configure your web server with a caching server like the Squid Proxy Server.

An example Squid-Apache Web Server setup could be:

1. Run the **Apache Web Server** on port **8080**
2. Run the **Squid** Proxy Server on port **80**
3. When a request from a client reaches port 80, the Squid Proxy Server will first check if it has the result stored. If so, Apache server will never be contacted to process the request thus reducing the server load.

To Configure the Squid Proxy Server in RHEL

The following example shows how you can configure the Squid Proxy Server for a site `www.foo.com` on a web server.

1. Change to the directory `/etc/opt/sun/webstack/squid`.
2. Open the `squid.conf` file in a text editor. Make the following changes to specify the port for the web site, and to specify the IP address for the web server. By default Squid Proxy Server listens to HTTP port 80. Also, specify access controls to allow access to your web site.

   ```
   http_port 80 accel defaultsite=www.foo.com
   cache_peer ip.of.server1 parent 80 0 no-query
   acl sites_server_1 dstdomain www.foo.com
   http_access allow sites_server_1
   ```
3. Create a cache directory by typing the following command.

   ```
   $ /opt/sun/webstack/squid/sbin/squid -z
   ```
4. Start Squid Proxy Server by typing the following command.

```
$ /opt/sun/webstack/squid/sbin/squid -NCd1
```

You can also start the Squid Proxy Server in RHEL as follows:

```
sudo /etc/init.d/sun-squid start
```

Note – See the logs at `/var/opt/sun/webstack/squid/logs` (especially in the cache.log file) to verify if there are any errors in configuring the Squid Proxy Server.

To configure the Squid Proxy Server as a Load Balancer in RHEL

Restart the Squid Proxy Server before you configure it as a load balancer.

This example uses two web servers. Each web server has two web sites. `www.foo.com` and `www.foo.net` are on Server 1, and `www.test.foo.com` and `www.test.foo.net` are on Server 2. This procedure explains how to configure the Squid Proxy Server as a load balancer for each site.

1. Change to the directory `/etc/opt/sun/webstack/squid`.
2. Open the `squid.conf` file in a text editor and make changes as follows.

```
cache_peer ip.of.server1 parent 80 0 no-query originserver name=server_1
acl sites_server_1 dstdomain www.foo.com www.foo.net
cache_peer_access server_1 allow sites_server_1
cache_peer ip.of.server2 parent 80 0 no-query originserver name=server_2
acl sites_server_2 dstdomain www.test.foo.com www.test.foo.net
cache_peer_access server_2 allow sites_server_2
cache_peer ip.of.server1 parent 80 0 no-query originserver round-robin
cache_peer ip.of.server2 parent 80 0 no-query originserver round-robin
```

Note – Round-robin algorithm does simple load balancing. The round-robin algorithm places all the jobs in a circular queue with a processing time slot for each job. The processor saves the status of a job after its time slot and resumes processing on its saved state in its next round of processing. This achieves virtual parallel-processing and load-balancing.

PHP Environment

If you have installed the Sun Web Stack software through the most common amp configuration option, you don't need to install the PHP runtime or extensions separately. But if you have not installed the complete amp stack, you can install PHP runtime by providing the php option during installation. For more information, see the Sun Web Stack Installation and Configuration Guide

For instance:

```
$ ./install php
```

Where is the PHP Interpreter Installed?

When you install PHP through the Sun Web Stack software, the PHP configuration and runtime files will be available under the locations provided in the following table:

Platform	Install Location
Solaris 10 OS	1. Installation Location - /opt/webstack/php/5.2/ 2. Runtime files - /var/opt/webstack/php/5.2/ 3. Configuration files - /etc/opt/webstack/php/5.2
Red Hat Enterprise Linux	1. Installation Location - /opt/sun/webstack/php/ 2. Runtime files - /var/opt/sun/webstack/php/5.2/ 3. Configuration files - /etc/opt/sun/webstack/php/5.2/

Available PHP Extensions

If you have installed the Sun Web Stack software using the amp installation option, some PHP modules are made available as extensions. Each of the modules has a respective INI file under the PHP conf.d directory. For instance, INI files for RHEL is available under /etc/opt/sun/webstack/php/5.2/conf.d directory. Most of these PHP extensions are enabled by default. Some custom third-party PHP extensions are not enabled by default. However, you can edit the PHP extension specific INI file for any specific configuration changes.

The following is the list of PHP extensions available through the Sun Web Stack software:

Bundled Extensions
`APC` (Enabled)
`dtrace` (Only for Solaris 10 OS)
`memcache` (Enabled)
`suhosin` (Disabled)
`tcpwrap` (Enabled)
`idn` (Enabled)
`xdebug` (Disabled)
`bz2` (Enabled)
`curl` (Enabled)
`ftp` (Enabled)
`gd` (Enabled)
`gettext` (Enabled)
`iconv` (Enabled)
`imap` (Enabled)
`ldap` (Enabled)
`mysqli` (Enabled)
`mysql` (Enabled)
`openssl` (Enabled)
`pdo` (Enabled)
`pdo_mysql` (Enabled)
`pdo_pgsql` (Enabled)
`pdo_sqlite` (Enabled)
`pgsql` (Enabled)
`sqlite` (Enabled)
`tidy` (Enabled)
`zlib` (Enabled)

Enabling/Disabling Bundled PHP Extensions

You can enable or disable a PHP extension by commenting out a line in the <extension>.ini available under the PHP conf.d directory.

Path to conf.d directory under Solaris 10 OS* - /etc/opt/webstack/php/5.2/.

Path to conf.d directory under RHEL* - /etc/opt/sun/webstack/php/5.2/.

For example, to enable the Suhosin PHP extension on RHEL, perform the following tasks:

1. Open /etc/opt/sun/webstack/php/5.2/conf.d/suhosin.ini in a text editor.
2. Un-comment the line enabling the Suhosin extension

```
extension=suhosin.so
```

To disable back the extension, add the comment character (;).

Note – You must restart the Apache Web Server for changes to take effect. For instance, sudo /etc/init.d/sun-apache22 restart.

Python Environment

Python is an object oriented scripting language and is available through the Sun Web Stack software on all supported platforms. Python scripts can be invoked through the Apache Web Server using mod_python. There are various web frameworks available for python including Django, Pylons, TurboGears, and Zope that are widely used by the community.

Note – Python is not provided through Sun Web Stack software for RHEL. You need to use the Python interpreter provided by the platform. Also mod_python is not part of the Sun Web Stack software.

You can download mod_python from http://httpd.apache.org/modules/pythondownload.cgi.

You can install Python by providing the python option during installation. For more information, see the Sun Web Stack Installation and Configuration Guide

For instance:

```
$ ./install python
```

Where is the Python Interpreter Installed?

When you install Python from the Sun Web Stack software, the Python runtime files will be available under the locations provided in the following table:

Platform	Install Location
Solaris 10 OS	Installation Location - `/opt/webstack/python/` Runtime files - `/var/opt/webstack/python`

Note – Python is **not available** through the Sun Web Stack software for RHEL.

Note – Python 2.5.2 has DTrace probes integrated. You can trace your Python-based applications for performance bottlenecks and other problems using the Solaris DTrace feature.

Ruby Environment

Ruby is available through the Sun Web Stack software only for Solaris 10 OS and OpenSolaris OS. The Sun Web Stack software does not include the Ruby interpretor on the Red Hat platform; use the OS-supplied Ruby instead.

You can install the Ruby runtime by providing the `ruby` option during installation. For more information, see the Sun Web Stack Installation and Configuration Guide

For instance:

```
$ ./install ruby
```

Where are the Ruby files?

When you install Ruby from the Sun Web Stack software, the Ruby configuration and runtime files will be available under the locations provided in the following table:

Platform	Install Location
Solaris 10 OS	Installation Location - `/opt/webstack/ruby/1.8` Runtime files - `/var/opt/webstack/ruby/1.8`

Note – Ruby is **not available** through the Sun Web Stack software for RHEL.

Tomcat Servlet Container

The Sun Web Stack software comes with the Tomcat 5.5.27 Servlet container for deploying JSP and Servlet based web applications. You can install the Tomcat Server by providing the `tomcat` option during installation. For more information, see the Sun Web Stack Installation and Configuration Guide

For instance:

```
$ ./install tomcat
```

In Solaris 10 OS*, Tomcat is available under `/opt/webstack/tomcat/5.5`.

In RHEL*, Tomcat is available under `/opt/sun/webstack/tomcat/5.5`.

To start the Tomcat Server in **Solaris 10 OS**, perform the following task:

1. Execute the following command:

```
/usr/sbin/svcadm enable sun-tomcat5
```

To stop the Tomcat Server, perform the following task:

```
/usr/sbin/svcadm disable sun-tomcat5
```

To start the Tomcat Server in **RHEL**, perform the following task:

1. Execute the following command:

```
/sbin/service sun-tomcat5 start
```

To stop the Tomcat Server, perform the following task:

```
/sbin/service sun-tomcat5 stop
```

NetBeans IDE as a Development Environment

This section shows you how to use the NetBeans IDE to build web applications using the Sun Web Stack components available through the Sun Web Stack software. First let us enumerate the stages involved in developing an end to end enterprise grade web application. In order to use the IDE for development, you will need the following:

- **Server support** - The IDE should support managing the Server life cycle through an intuitive and easyuse interface.
- **Languages Support** - The IDE should support working with the programming language or web application framework that your web application will be eventually built on.
- **Database support** - The IDE should be well integrated with the development, testing and production database.

Typically, your development environment should enable you to build web applications with minimal effort, offering you extensive support for stack level configuration through property editors, wizards, and widgets. Server support, languages support, and database support are the three prime facets that any IDE can offer for web development.

The NetBeans IDE takes care of all these factors with an elegant interface thereby making web development easy. You can build your web applications in PHP or use Rails framework and work with databases like MySQL or PostgreSQL in the NetBeans IDE.

Note – If you are already using the NetBeans IDE, you can use the Sun Web Stack components along with NetBeans IDE by following some steps. For instance, you can use **Add Server** option of NetBeans IDE to add the Apache Tomcat Server 5.5 provided through the Sun Web Stack software.

You can also use the NetBeans IDE for Java/PHP/Ruby development. The NetBeans IDE supports integrated configuration windows for managing web servers and databases directly from the IDE. Note that the version of NetBeans available in the official pkg repository may be of an older version than the one available through the NetBeans web site. The following sections show how to configure the NetBeans IDE 6.0 for your development needs.

The NetBeans IDE supports various Java EE servers and other web servers for application deployment. You can have multiple servers configured in NetBeans irrespective of the server that will be used in your projects. Thus you can have the Apache 2 Web Server configured for deploying PHP files, the Apache Tomcat container configured for deploying Java based web applications or a WEBrick Server for deploying Rails applications.

Apache Web Server

To configure the Apache Web Server in the NetBeans IDE 6.0, perform the following tasks:

1. In NetBeans IDE, go to the Services Tab.
2. Expand the Web Servers node.
3. Right-click the Web Servers node and click Add Web Server.
4. Choose Server Type - Add a connection name. For example, type Apache 2. Select Local Web Server with File Access option so that you can instruct the NetBeans IDE to use the Apache Web Server installed through the Sun Web Stack software.
5. Find Configured Hosts - Select Auto Configuration. Click Find Servers. The Server Location field is automatically populated.
6. Configure HTTP Server - Enter Server Name as `localhost`.
7. Configure File Access - Enter Document Root. For Solaris 10 OS the Document Root is `/var/opt/webstack/apache2/2.2/htdocs`. For RHEL the Document Root is `/var/opt/sun/webstack/apache2/2.2/htdocs`.

After adding the server, check if the Apache Web Server shows up in the Web Servers list by clicking the Services tab.

These are the basic steps to get you started with using the Apache Web Server for deploying your PHP based web applications. Since the Apache Web Server is already configured to support PHP files, you do not need to perform any other configuration apart from getting and installing the PHP plug-in.

Note – The procedure mentioned above is applicable only for the NetBeans IDE 6.0. If you are using NetBeans IDE 6.1 or higher, you do not need to configure web servers up front. You can configure a server while creating projects.

Tomcat Web Server

The Sun Web Stack software comes with a Tomcat 5.5 Servlet container for deploying your JSP based web applications.

To configure Tomcat Web Server in the NetBeans IDE:

1. In NetBeans IDE, go to Tools > Servers.
2. In the left pane, click Add Server.
3. Choose Tomcat 5.5 from the servers list.
4. In the Catalina Home field, type `/opt/sun/webstack/tomcat/5.5` (Red Hat) or `/opt/webstack/tomcat/5.5` (Solaris 10 OS).

5. Add the user name and password for the manager role as provided in the `tomcat-users.xml` file in the Tomcat configuration directory. You can also let NetBeans create a new user if the user entered in the user name text field does not exist.

When Tomcat is successfully configured in the NetBeans IDE, the Servers node shows Tomcat 5.5 entry.

You can right-click the Tomcat 5.5 node and select Start to start the server.

For PHP Development

PHP is an interpreted, dynamic, web page scripting language. NetBeans IDE 6.0 has support for PHP editing and debugging through an add-on component called PHP plug-in. The plug-in supports PHP5 and the following features:

When you have downloaded and configured the NetBeans plug-in, you should be able to create new PHP projects. However, to run and debug your PHP projects, you will need:

- **Apache Web Server** - Available through Web Stack. You can use a local server or a remote server with FTP access. Typically a local web server is utilized for development and debugging while the production environment is located on a remote web server.
- **PHP engine** - Available through Web Stack.
- **PHP debugger** - Available through Web Stack PHP Extension. The NetBeans 6.0 IDE PHP Plug-in allows you to use XDebug, but using a debugger is optional.

For using PHP and Apache 2 through the Sun Web Stack software, you do not need any additional configuration.

Note – PHP language interpreter is integrated into the Apache Web Server. PHP module with MySQL and PostgreSQL database support is provided for the Apache Web Server. If you need to configure PHP with Apache 2.2 worker MPM, then you need to use Apache 2.2 Web Server + PHP FastCGI.

Some PHP modules are packaged with the Sun Web Stack software as extensions. Each of the modules has a respective INI file under `/etc/opt/sun/webstack/php/5.2/conf.d` directory in **RHEL** or `/etc/opt/webstack/php/5.2/conf.d` directory in **Solaris 10 OS**. These PHP extensions are enabled by default. Some custom third-party PHP extensions are not enabled by default. However, you can edit the PHP extension specific INI file for any specific configuration changes.

For Ruby/Rails Development

The NetBeans IDE features plug-ins that provides an integrated development environment for building, running, testing, and debugging Ruby and Ruby on Rails applications. NetBeans supports Ruby project types including Ruby files, RSpec specification files, and YAML files. You can configure NetBeans to use the native ruby available through Web Stack. The IDE also supports Rake build tool integration and gives you access to the Interactive Ruby Shell.

Note – If you need to use the Gem Manager, you must have gem installed, and the IDE must have write permission to your Ruby installation directory.

In Solaris 10 OS, `/var/opt/webstack/ruby/1.8/gem_home` contains the Ruby gems repository. Configure the `GEM_HOME` environment variable to point to `/var/opt/webstack/ruby/1.8/gem_home` to use this installation of Ruby gems. `/var/opt/webstack/ruby/1.8/bin` contains the Ruby executable as well as other utility programs. These programs are linked from `/usr/bin`.

Before proceeding to use NetBeans for Ruby, set the environmental variable `GEM_HOME`.

- For sh - `export GEM_HOME=/var/opt/webstack/ruby/1.8/gem_home`
- For csh - `setenv GEM_HOME /var/opt/webstack/ruby/1.8/gem_home`

Note – To use gem, make sure you have direct access to the Internet. If your system is behind a firewall or if it uses a proxy server, set the `HTTP_PROXY` variable.

If you do not set the `GEM_HOME` environmental variable, the NetBeans IDE will not work because the available gems cannot be enumerated if you are using a native Ruby environment. For this reason you need to set `GEM_HOME` in order for the gem repository to be accessible from the NetBeans IDE. Also the directory referenced by `GEM_HOME` should be writable. So make sure you execute the following command:

```
$ chmod -R a+w /var/opt/webstack/ruby/1.8/gem_home
```

Ruby on Rails (RoR)

Rails is a full-stack framework for developing database-backed web applications. Rails provide a pure-Ruby development environment. Ruby gems is the standard Ruby package manager. If you have successfully installed Ruby through the Sun Web Stack software, Ruby gems are already loaded. Hence you can install Rails and its dependencies by executing the following command:

```
/usr/bin/gem install rails --include-dependencies
```

You can also install rails directly from the NetBeans IDE rather than invoking the gem command through a command line. The Ruby on Rails environment is now ready for development. You can now use the NetBeans IDE to create a Ruby on Rails project. When you create a project, the IDE creates the same folders and files that a rails command would create.

Ruby Gems

Ruby Gems is a system for managing Ruby software libraries. Ruby code packaged in this manner is called a gem. Gems enable downloading, installing, and managing your Ruby libraries. The Web Stack package in Solaris does not include a lot of packaged gems because you can use Ruby Gems to download and install additional gems. You can use the the NetBeans interface for managing Gems.

For more information on configuring Ruby Gems, refer to the documents at the NetBeans Ruby site.

Deploying Common Open Source Applications

This section shows you how you can deploy some of the common web applications that you see today. The steps involved includes the Apache Web Server and MySQL DB configuration.

Drupal

Drupal is an open-sourced Content Management System (CMS) that can be used to host a web-site with pluggable components like blogs, forums, posts, feeds, editorials, and triggers. Drupal is one of the common applications that a hosting service provider will offer you with a control panel interface for installing and configuring the application. Take for instance, cPanel that runs Fantastico. You can install and configure your blogs/websites and other applications through an intuitive interface.

If your hosting provider has no support for interfaces like these, you may need to install and configure applications manually.

What do you need?

1. **PHP Runtime Support** - Drupal is a PHP based application, so you need a server with PHP support. If your hosting provider has already configured the Apache Web Server with PHP, you are already set.
2. **MySQL DB Support** - Drupal needs a database to store the website content. You need to install and start the MySQL server.

To install and configure Drupal, perform the following steps:

1. **Get Drupal** - You can download the Drupal application from http://drupal.org/download.
2. **Get the Sun Web Stack software** - Refer the earlier sections for setting up AMP.
3. Extract Drupal package and move the directory to Apache's htdocs directory. In Solaris 10 OS, the path is `/var/opt/webstack/apache2/2.2/htdocs`. In RHEL, the htdocs directory is available under `/var/opt/sun/webstack/apache2/2.2/htdocs`
4. Set necessary permission for installing Drupal - From the Drupal directory, run `chmod o+w sites/default/`
5. Create a Drupal DB - Run `mysqladmin -u root -p create drupal` (Provide the password for root. If you have not changed the default password for root, it is an empty string).
6. Set necessary permission for Drupal DB - Run `mysql -u root -p`. After providing the password, execute the query - `grant select, insert, update, delete, create, drop, index, alter on drupal.* to 'root'@'localhost' IDENTIFIED BY '';` from the MySQL prompt.
7. Access the URL - http://localhost/drupal to proceed with the installation.

Note – You can also install Drupal directly from the Web Stack repository, so you don't have to configure the system before installation.

Geeklog

Geeklog is a web content management system suitable for running full-featured community sites. It supports article posting, threaded comments, event scheduling, and link management.

Geeklog is widely used by the community primarily because of the ease of installation. If your hosting provider is ruuning Fantastico, you can install and configure Geeklog from the cPanel interface.

To install Geeklog, perform the following tasks:

1. Get the Sun Web Stack software.
2. Get Geeklog - You can download Geeklog from the Geeklog site.
3. Extract the archive.
4. Rename the publicc_html sub-directory to geeklog and move this directory to `/var/opt/webstack/apache2/2.2/htdocs` folder for Solaris 10 OS or `/var/opt/sun/webstack/apache2/2.2/htdocs` for RHEL.
5. Move the rest of the content (except publichtml sub-directory) to `htdocs` as a separate folder with the name geeklog_files.

6. Create Geeklog DB - Execute `mysqladmin -u root -p create geeklog` to create a DB by name geeklog. You will be prompted to provide the password for root. The default password for root is an empty string.
7. Set necessary permission for Geeklog DB - Run `mysql -u root -p`. After providing the password, execute the query - `grant select, insert, update, delete, create, drop, index, alter on geeklog.* to 'root'@'localhost' IDENTIFIED BY '';` from the MySQL prompt.
8. (Optional) Edit `htdocs/geeklog_files/db-config.php` file to provide the MySQL connection information as shown below:

```
$_DB_host = 'localhost';
$_DB_name = 'geeklog';
$_DB_user = 'root';
$_DB_pass = '';
$_DB_table_prefix = 'gl_';
$_DB_dbms = 'mysql';
```

9. Set necessary permission for installing Geeklog - From the htdocs directory, run `chmod o+w geeklog` and `chmod o+w geeklog_files`
10. Invoke the installer script - From a browser window, invoke http://localhost/geeklog/admin/install/index.php. At this point you will get an error stating db-config.php is not found.
11. Provide the path in the text field as `/var/opt/webstack/apache2/2.2/htdocs/geeklog_files/db-config.php` (For Solaris 10 OS). `/var/opt/sun/webstack/apache2/2.2/htdocs/geeklog_files/db-config.php` (For RHEL).
12. Follow the installation steps. You may need to provide DB related information.
13. Remove the geeklog_files directory from the htdocs directory.

For more info, read the Geeklog installation document.

If you want to migrate your content to Drupal later, read this document.

Geeklog can easily be extended by the many community developed plugins to alter its functionality. Available plugins include forums, image galleries, and many more.

More Applications

See Web Stack Documentation Blog for more examples.

More information about the Sun Web Stack Software

- Getting Started with Sun Web Stack integrated with OpenSolaris 2008.11
- Using Memcached with Solaris Zones
- DTrace and Sun Web Stack
- Using lighttpd and Apache Web Server