Web and Application Services Overview

Open Enterprise Server 2 SP3

May 3, 2013

Novell

Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

Novell Open Enterprise Server (OES) 2 includes a collection of open source and Novell products that let you build, deploy, host, and use Web sites and Web applications that speed up business processes without jeopardizing the security of business information.

The guide is divided into the following sections:

Audience

This guide introduces you to Web and application services, and explains how you can begin using them to meet the demands of your business. It is intended for Web or network administrators who install and manage Web site content and applications. Developers who write Web-based applications to run in the OES environment might also find the information in this overview helpful.

Feedback

We want to hear your comments and suggestions about this guide and the other documentation included with Novell OES. Please use the User Comment feature at the bottom of each page of the OES online documentation.

Documentation Updates

For the most recent documentation, visit the OES 2 Documentation Web site (http://www.novell.com/documentation/oes2).

Additional Documentation

Each Linux component discussed in this overview has its own documentation on the Web. For details about how to configure and manage each component, refer to the following documentation:

- Apache 2.2 Documentation (http://httpd.apache.org/docs/2.2/)
- Tomcat 4 documentation (http://jakarta.apache.org/tomcat/tomcat-4.1-doc)
- MySQL documentation (http://dev.mysql.com/doc)

For information about Novell exteNd, see the Novell exteNd documentation Web site (http://www.novell.com/documentation-index/index.jsp?category=exteNd).

1 OES Web and Application Services Overview

Novell Open Enterprise Server (OES) 2 includes a collection of open source and Novell products that let you build, deploy, host, and use Web sites and Web applications that speed up business processes without jeopardizing the security of business information. Using OES, you can use the full range of Web and application services.

This section covers the following topics:

1.1 Introduction to Web and Application Services

The rise of the Internet and the World Wide Web sparked a revolution not only in network communications but also in application design and development. Programmers have encapsulated pieces of business functionality into distinct objects or components, and then made them available as self-contained Web services that can be accessed using Internet-based protocols and tools.

As network servers have become capable of supporting Internet-based services, software developers have devised new programming paradigms to take advantage of the widespread availability of these services. This new class of software is categorized as Web-based or Web-enabled applications.

This section introduces some basic concepts and technologies that are helpful to understand when working with Web services and Web applications.

1.1.1 What Are Web Services?

The term *Web services* can be confusing because it is used in many different ways. In most contexts, Web services are business logic components that can be connected together and exchange data to perform a useful task. The components can be internal or external to an organization, and they

communicate using Internet-based protocols such as the HyperText Transfer Protocol (HTTP). In brief, Web services run on servers and process substantial amounts of data that users want to be able to access quickly and easily.

A popular programming model in which individual Web services are combined to create a functional whole is the *service-oriented architecture*. In this model, a service consumer sends requests to a service provider over a standard connection. The request and subsequent response are defined in a way that is understandable to both the consumer and provider.

Most Web services use Extensible Markup Language (XML) to define the format of request and response messages. XML features a tagged structure that provides the needed flexibility for exchanging data between disparate components. XML can also be used to define how data is stored in a database.

Simple Object Access Protocol (SOAP) provides a standard for enveloping and sending Web services messages. It is an XML messaging specification that describes a message format along with rules for exchanging data in the proper sequence between structured data types and arrays. SOAP generally uses HTTP, but it can use other standard Web protocols as well.

In the service-oriented architecture, service consumers can find available service providers through various discovery mechanisms. One such mechanism is the Universal Description, Discovery, and Integration (UDDI) registry. As Web services are developed, they can be added to the UDDI registry. The registry can then be searched in various ways to find the Web services available for a particular organization and obtain contact information.

1.1.2 What Are Web Applications?

In its simplest form, a Web application is an interactive system that allows its users to execute business logic that resides on a server and to view the results of that logic through a Web browser on a client workstation. The defining factor that makes the system a Web application is that the server and client communicate over the Internet. In brief, Web applications make the data processed by Web services available to users quickly and easily through their Web browsers.

Web applications are built on a client/server architecture. The business logic is contained in the application itself, which runs on a Web server and uses HTTP to communicate with clients over the Internet. The Web server manages the application, passes requests from clients to the application, and returns the application's responses to the client.

On the client side, the Web application is viewed with a browser. The application's user interface takes the form of HyperText Markup Language (HTML) pages that are interpreted and displayed by the browser. In addition to text, these HTML pages can contain Web forms, image files, audio and video clips, and other types of displayable data.

Although Web applications can use a Web site as the front end to their business logic, you can do many things in a Web application that you can't do with a static Web site, such as:

- Identify specific users and present a customized interface for each user
- Collect information from users and store that information on the server
- Perform tasks for users, such as retrieving information from a database, registering to access specific content, or placing an order for a product

1.1.3 Web Application Tools (Java and J2EE)

Java has become a standard programming language for Web applications because it is simple and portable to various hardware platforms. All you need to run Java applications is a Java Virtual Machine (JVM) for your particular platform. JVMs are available for almost every server platform in existence, including SUSE Linux Enterprise Server 10, Novell NetWare, Sun Solaris, Microsoft Windows, and Apple Macintosh OS.

Java 2 Platform, Enterprise Edition (J2EE) is a widely used environment for developing enterprise Web applications. J2EE offers a multitiered distributed application model, the ability to reuse components, integrated XML-based data interchange, a unified security model, and flexible transaction control. Best of all, applications developed for a J2EE application server are not tied to any one vendor's products or APIs.

The J2EE specification defines the following components:

- **Servlets:** A Java servlet is a server-side component that provides a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems. A servlet dynamically processes client requests and constructs responses. Servlets are commonly used to process forms, handle redirects or authenticate user names and passwords, and create dynamic content for a Web application.
- **JavaServer Pages:** JavaServer Pages (JSPs) are text-based documents that execute as servlets but allow a more natural approach to creating Web content. JSPs allow Web developers to rapidly develop and easily maintain dynamic Web pages that leverage existing business systems. JSP technology separates the user interface from content generation, enabling the overall page layout to be changed without altering the underlying dynamic content.
- **Enterprise JavaBeans:** Enterprise JavaBeans (EJBs) are the basic components of an architecture that allows developers to create objects that precisely model the structure and logic of a business application domain. The system-level details of building the distributed application are abstracted out, enabling domain experts to be developers who freely focus on solving business problems. EJB technology enables rapid development of distributed, transactional, secure, and portable Java-based applications.

1.1.4 Enabling Technologies

Web applications employ various enabling technologies to make their content dynamic and to create user interfaces into the business logic on the server.

- "Scripting Languages" on page 9
- "Servlet Containers" on page 10
- "Web Database Servers" on page 10
- "Application Servers" on page 10

Scripting Languages

Foremost among the enabling technologies are scripting languages such as PHP and Perl.

PHP (PHP: Hypertext Preprocessor) is a powerful server-side scripting language that is easy to learn. It offers all of the power and flexibility of JSP, but does not require as much memory and processing power. You mix specially delimited PHP code in with regular HTML to create a dynamic Web page. PHP is commonly used to access Web databases such as MySQL. It also supports library extensions to leverage standard services such as LDAP, FTP, POP3, Java, and many others.

Perl (Practical Extraction and Report Language) is another server-side scripting language commonly used by Web programmers to create scripts for Web servers. It uses a syntax similar to C/C++ and its file-manipulation and text-manipulation facilities make it ideal for tasks involving software tools, database access, graphical programming, networking, and system management.

Servlet Containers

A complementary component for both servlets and JSPs is the *servlet container*. The container acts as a simple application server that executes Java servlets and renders Web pages that include JSP code. It provides necessary functions such as life cycle management and interaction with a Web server.

The official reference implementation of the Java servlet API is Jakarta-Tomcat, an open source project released under the Apache Software Foundation. Tomcat is typically used in conjunction with a Web server such as Apache.

Web Database Servers

MySQL is an open source, structured query language (SQL) Web database server that is often used by PHP and Perl developers because its syntax is similar to those languages. It offers fast performance and is designed to work well with Web servers. It is widely used in building basic database-driven Web applications.

PostgreSQL is another Web database server that offers more advanced features often found in commercial database systems, such as transactions, subselects, triggers, views, referential integrity, and sophisticated locking. It is often used to provide more complex database functionality for Web sites and Web applications.

Application Servers

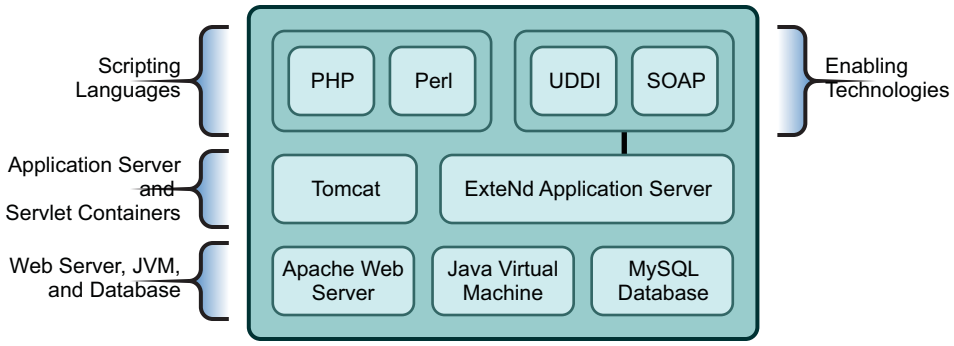
In more sophisticated Web application models, an application server is added to enable the system to manage business logic and track the user's progress through the application. The application server software runs in a middle tier, between Web browser-based clients and back-end databases and business applications. The application server handles all of the application logic and connectivity that old-style client/server applications contained.

Examples of J2EE application servers are the open source JBoss application server and the commercial Novell exteNd Application Server.

1.1.5 General Web and Application Services Architecture

The following diagram shows the basic architecture of the Web components and services that are commonly used to host Web sites and build Web applications.

Figure 1-1 *Architecture of Key Web Components and Technologies*

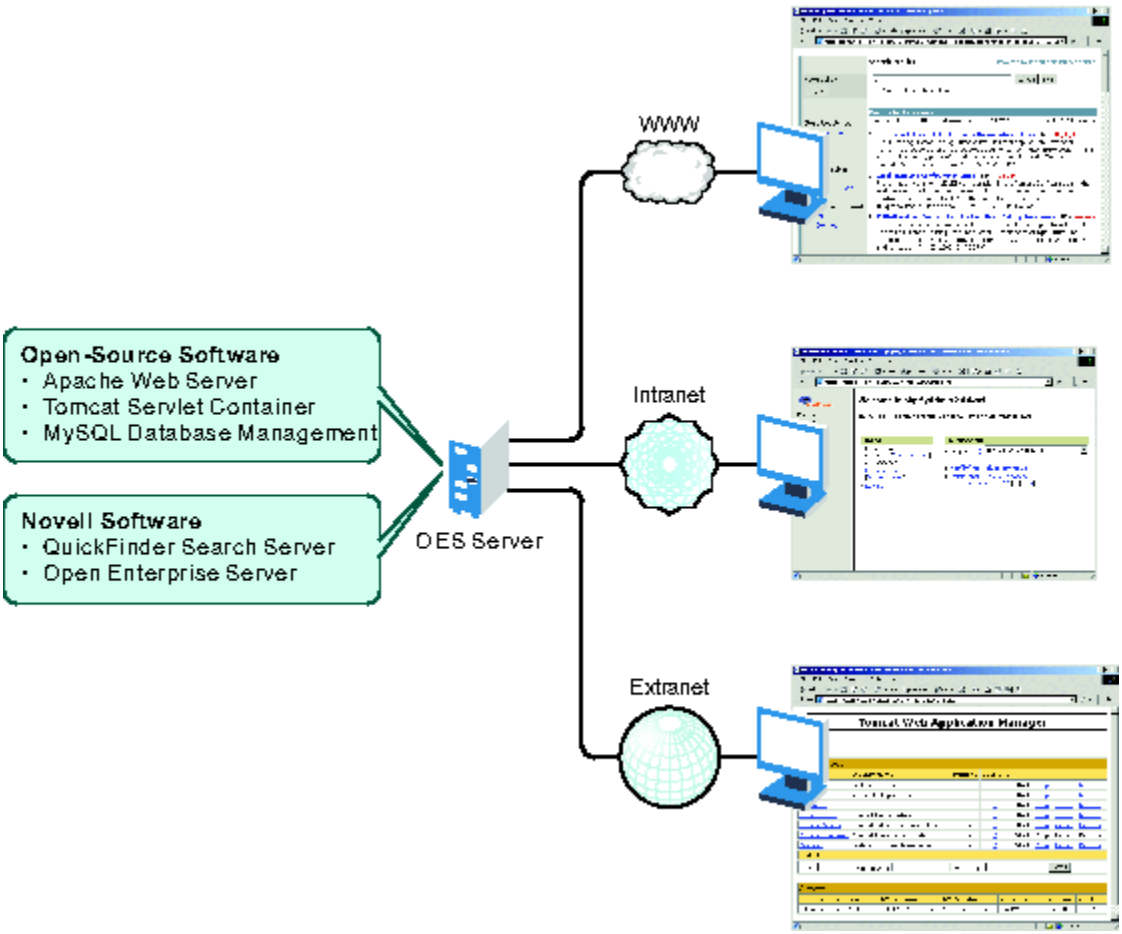


1.2 OES Components That Provide Web and Application Services

OES comes bundled with all of the Web and application services components you need to host dynamic Web content and deploy Web applications that you can either build yourself or download from the World Wide Web. Some of these components are developed by the open source software community, while others are developed by Novell. Each component offers an important building block that lets you build the solutions that best meet your business needs.

The following diagram illustrates how you can combine open source software and Novell software to provide Web-based business solutions for employees, customers, and partners.

Figure 1-2 *Open Source and Custom Built Solutions*



With the Web components available in OES, you can:

- Host multiple Web sites on a single OES server.
- Manage all instances of the Apache Web server from one interface using Apache Manager (regardless of what platform they are running on in your network).
- Choose from hundreds of free Web applications that can be downloaded from the Internet and run on your OES server.
- Build and host your own Web database applications.
- Choose from popular scripting languages to build your own dynamic Web content.

- Build powerful Web applications and services using the JBoss or Novell exteNd Application Server, which includes SOAP and UDDI components, as well as rapid application development support and application deployment capabilities.
- Add search and print functionality to any Web site, anywhere on the World Wide Web or on a company intranet.

Some of the key benefits OES has to offer in the area of Web and applications services include the following:

- Open source components that help you steer away from vendor lock-in and proprietary solutions. Applications that you develop can run on any other J2EE compliant platform, including UNIX and Windows operating systems.
- Valuable services for end users that enhance personal and team productivity.
- A strong J2EE and open source development model.
- A broad range of industry standard API sets.
- A broad selection of development tools and deployment models for developers. This provides tremendous flexibility in those cases where IT organizations decide to repurpose their servers.
- Lower IT spending because open source products are free and platform independent.

The following sections introduce each Web and application services component included with OES:

- Section 1.2.1, "Web Hosting: Apache Web Server 2.0," on page 13
- Section 1.2.2, "Servlet Support: Tomcat Servlet Container," on page 14
- Section 1.2.3, "Scripting: PHP and Perl," on page 14
- Section 1.2.4, "Web Databases: MySQL," on page 14
- Section 1.2.5, "Custom Web/J2EE Application: JBoss," on page 15
- Section 1.2.6, "Web and Network Search Capability: QuickFinder Server," on page 15

1.2.1 Web Hosting: Apache Web Server 2.0

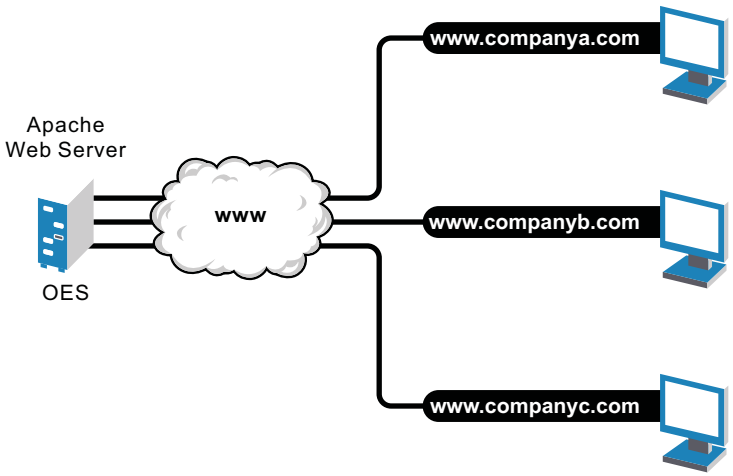
Apache is the most popular Web server being used on the World Wide Web today. Its popularity comes from the fact that it is the most reliable and secure Web server available. It runs on all major platforms, is capable of hosting even the most complex Web sites, and can scale to handle thousands of simultaneous connections.

The Apache Web Server 2.0 serves as the foundation Web server upon which you can build Web sites and host Web applications for use in your business.

Key uses and benefits of using Apache in OES include the following:

- It provides a highly reliable and fast Web server for hosting simple or complex Web sites.
- It is preconfigured to work with Jakarta-Tomcat, the servlet container created by the Apache Foundation, which can be used to host servlets and JavaServer Pages (JSPs) for automating business processes.
- It is ideal for Web application development and testing.
- It lets you set up multiple virtual hosts for hosting multiple Web sites (with their own domain names) all from a single installation of Apache.

Figure 1-3 *Apache Running on an OES Server and Hosting Multiple Web Site*



OES includes Apache Web Server 2.0 for Linux. It features a hybrid multi-process/multi-threaded implementation, filtering, simplified configuration, and a new API, along with extension modules to support Secure Sockets Layer (SSL), LDAP authentication, and multi-language error messages.

1.2.2 Servlet Support: Tomcat Servlet Container

OES includes a Jakarta-Tomcat container for Linux. Tomcat is ideal for running basic Java servlet and JSP applications. OES also includes Tomcat 5 for Linux, which implements the Java Servlet 2.4 and JSP 2.0 specifications.

If you are relatively new to, or inexperienced with, Java programming and do not plan to build more advanced J2EE applications, the Tomcat container should satisfy your needs. It is very stable and includes all of the features of a commercial Web application container.

1.2.3 Scripting: PHP and Perl

Scripting languages and visual builder tools have gained popularity in recent years because of their ease of use in delivering content to the Web. OES provides a choice of scripting languages and the engines to run them. You can use these tools to develop Web applications and administration utilities.

The scripting technologies integrated with OES Linux include industry standard PHP and Perl.

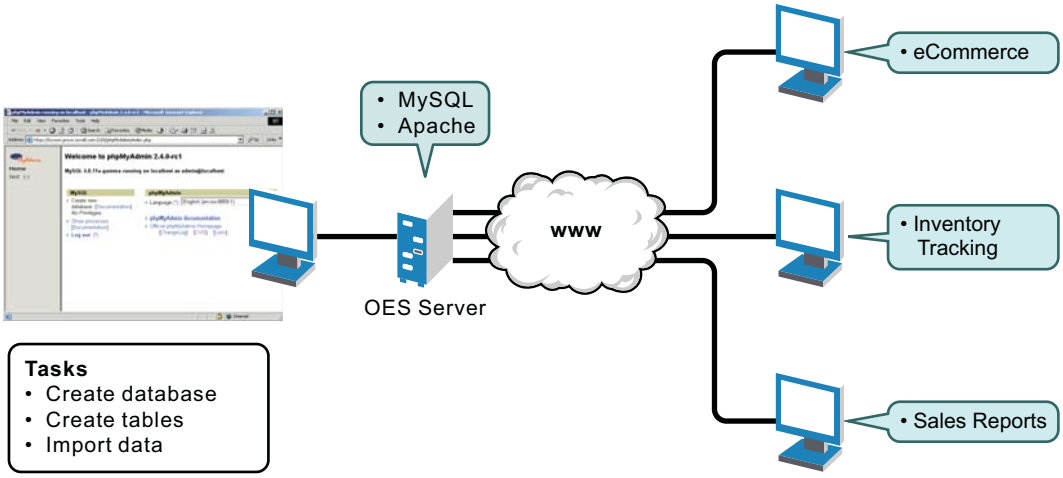
1.2.4 Web Databases: MySQL

OES includes the open source MySQL database server on the Linux platform. When combined with a Web application and a Web server, MySQL is a very reliable and scalable database for use in hosting eCommerce and business-to-business Web applications.

To manage your MySQL database, you can use the open source phpMyAdmin application written in the PHP language that provides a Web-based administration tool.

The following diagram shows how MySQL can be used to host Web database applications such as eCommerce or inventory tracking.

Figure 1-4 *MySQL and phpMyAdmin: Hosting Several Web Database Applications*



NOTE: The more powerful PostgreSQL database server comes with SUSE Linux Enterprise Server 9 and later.

1.2.5 Custom Web/J2EE Application: JBoss

When you need greater processing power beyond what scripting or Web application hosting with Tomcat can offer, OES offers a J2EE-certified application servers: JBoss. Bundled with SLES 10, JBoss provides enterprise-class security, transaction support, resource management, load balancing, and clustering. JBoss application server is a comprehensive, J2EE-certified platform for building and deploying enterprise-class Web applications. It supports JSP, EJBs, and all other standard J2EE components and technologies.

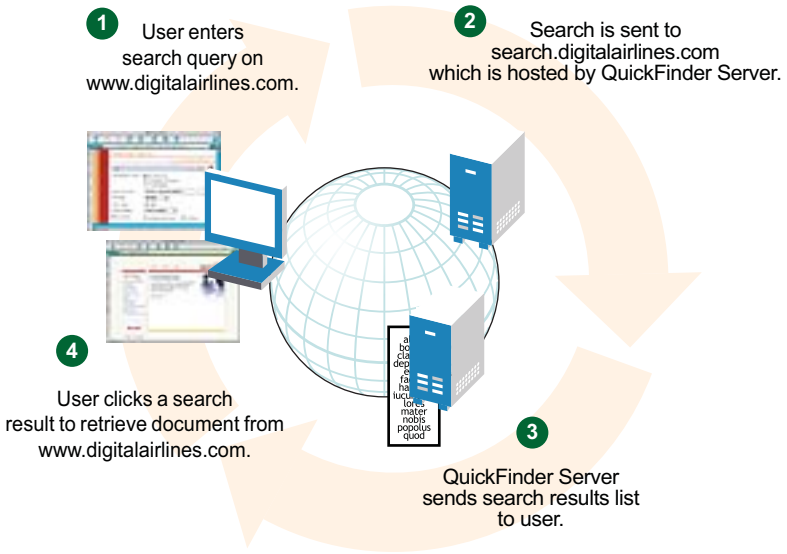
1.2.6 Web and Network Search Capability: QuickFinder Server

No Web solution is complete without capable searching functionality that provides users with a method for finding information they need, when they need it. That is why OES includes the Novell QuickFinder Server on the Linux platform.

Novell QuickFinder Server lets you add search and print functionality to any Web site, anywhere on the World Wide Web or on a company intranet. You can use it on your own enterprise-wide Web site or to host search services for business partners or clients.

Visitors to your Web or intranet site enter search terms in the search form that you place on the pages of your Web site. The search term is used to find matches contained in indexes you create using the QuickFinder Server Manager, a Web-based management utility. Search results, including matching URLs, are sent back to the user's Web browser.

Figure 1-5 *How QuickFinder Server Handles a User's Search Query*



1.3 What's Next

- To learn more about developing Web applications for the OES environment, see the Novell Developer Web site (http://developer.novell.com).
- For general OES installation instructions for Linux, see the *OES 2 SP3: Installation Guide*.

2 What's New or Changed for Web Services

This section describes enhancements and changes in Novell Cluster Services since the initial release of version 2.0 in Novell Open Enterprise Server (OES) 2.

2.1 What's New (May 2013)

Upgrade to eDirectory 8.8.7

An upgrade to Novell eDirectory 8.8 SP7 is available in the April 2013 Scheduled Maintenance for OES 2 SP3. For information about the eDirectory upgrade, see TID 7011599 (http://www.novell.com/support/kb/doc.php?id=7011599) in the Novell Knowledgebase.

There will be no further eDirectory 8.8 SP6 patches for the OES platform. Previous patches for Novell eDirectory 8.8 SP6 are available on Novell Patch Finder (http://download.novell.com/patch/finder/#familyId=112&productId=29503).

3 Configuring Apache HTTP Server on OES Servers with Novell Cluster Services

The Apache HTTP Server is an open source Web server developed by the Apache Software Foundation (http://www.apache.org). On a Novell Open Enterprise Server (OES) 2 SP3 cluster, you can use Novell Cluster Services to cluster the Web content for your personalized Web sites. The Apache service is not cluster aware and must run on each server in the cluster.

Clustering your Web site content helps make your Web site highly available for your customers. With Novell Cluster Services, if your Web server fails, you can use any of the servers in the cluster to host your Web site, which results in virtually zero down-time for your customers.

This section describes key considerations for configuring the Apache virtual hosts for your personalized Web sites.

3.1 Prerequisites for Using Apache on OES Servers

The following setup is required to reuse the Apache cluster resources from your NetWare cluster:

- When you install OES services on the server, Novell-ready versions of Apache 2 (64-bit) and Tomcat 5 are automatically installed and configured. You manually manage Apache services with the Apache configuration files. Use a text editor to create or modify the configuration files, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes.

  WARNING: Do not install the Linux *Web and LAMP* pattern. Do not use the *HTTP Server* option in YaST to configure Apache or virtual host settings on an OES server. It overwrites essential OES settings for Apache and breaks the existing setup. For recovery information, see Section 3.3.1, "Apache Server Errors after Using the HTTP Server Option in YaST," on page 33.

- You can use the Novell Storage Services (NSS) file system or Linux file systems to host your Web content:
  - **NSS volumes:** Install the *Novell Storage Services* pattern on each OES node in the cluster. For information, see "Installing and Configuring Novell Storage Services" in the *OES 2 SP3: NSS File System Administration Guide for Linux*.

Novell Cluster Services supports cluster resources for NSS pools. For information about creating clustered NSS pools and volumes, see "Configuring Cluster Resources for Shared NSS Pools and Volumes" in the *OES 2 SP3: Novell Cluster Services 1.8.8 Administration Guide for Linux*.

- **Linux volumes:** Linux POSIX file systems (such as Btrfs, Ext2, Ext3, Reiser, and XFS) are installed automatically.

  To use NCP-enabled Linux volumes, install NCP Services and Novell Remote Manager on each OES node in the cluster. For information, see the *OES 2 SP3: NCP Server for Linux Administration Guide*.

  Novell Cluster Services supports cluster resources for shared Linux POSIX volumes on . For information about creating clustered Linux POSIX volumes, see "Configuring and Managing Cluster Resources for Shared Linux POSIX Volumes" in the *OES 2 SP3: Novell Cluster Services 1.8.8 Administration Guide for Linux*. For information about enabling NCP on the volume, see "Configuring NCP Volumes with Novell Cluster Services" in the *OES 2 SP3: NCP Server for Linux Administration Guide*.

- You can host multiple Web sites on the same server. You must configure an Apache virtual host for each Web site. In a Novell Cluster Services cluster, you must configure an Apache virtual host on one OES node, then copy the configuration files to every OES node in the cluster. For information, see Section 3.2.3, "Creating and Configuring a Virtual Host for Each Web Site," on page 23.
- In a Novell Cluster Services cluster, the directories you specify in the `DocumentRoot` directive and any `Alias` directives for a virtual host should reside on the same pool cluster resource so they can fail over together. The location that contains the Web content should be a directory on the volume, not the root of the volume. Specify the full Linux path of the directory. Linux paths are case-sensitive.

  For example, the full Linux path of the `/www/mysite` path on an NSS volume `APACHEVOL` is

  ```
  /media/nss/APACHEVOL/www/mysite
  ```

  For Linux POSIX volumes, the `/www/mysite` path should be added to the volume's mount point path, such as:

  ```
  /mnt/apachevol/www/mysite
  ```

- The following permissions are required:
  - The user `wwwrun` must be the file owner of the Web site directories and files. The group can be the system `root` or the Apache group `www`.
  - If Web content resides on an NSS volume or an NCP-enabled Linux volume, the following additional permissions are required:
    - Enable the eDirectory user `wwwrun` and group `www` with Linux User Management (LUM). OES automatically creates and LUM-enables the user and group when you install the first OES server in a Novell eDirectory tree.
    - Assign the eDirectory user `wwwrun` as a file system trustee with Read and File Scan rights for the directory you specify in the `DocumentRoot` directive in the virtual host configuration file.

For information about the default OES setup for Apache and setting up virtual hosts, see Section 3.2, "Using Apache HTTP Server on OES Servers," on page 21.

3.2 Using Apache HTTP Server on OES Servers

When you set up OES services on the server, Novell-ready versions of Apache 2 HTTP Server software and Tomcat 5 are automatically installed. Apache and the OES Welcome Web site are automatically configured for non-secure port 80 and secure port 443. The Apache HTTP Server daemon (`httpd2`) starts automatically on server restart.

To set up personalized Web sites, you must manually create a virtual host configuration file for each Web site. Templates for secure SSL virtual host and non-secure virtual host configuration files are available in the `/etc/apache2/vhosts.d/` directory. Use a text editor to create or modify the configuration files, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes.

WARNING: Do not use the *HTTP Server* option in YaST to configure Apache or virtual host settings on an OES server. It overwrites essential OES settings for Apache and breaks the existing setup. For recovery information, see Section 3.3.1, "Apache Server Errors after Using the HTTP Server Option in YaST," on page 33.

3.2.1 Understanding the Default OES Setup of Apache HTTP Server

When you install services from the OES Add-On disk, the following Apache setup is configured:

Apache and Tomcat Installation

Novell-ready versions of Apache 2 HTTP Server software and Tomcat 5 are automatically installed when you set up OES services on a server. OES installs the Apache `prefork` mode or `worker` mode packages, depending on the OES services you install. If OES installs Apache Prefork packages, Apache should run in `prefork` mode rather than `worker` mode. OES sets the preference for Prefork mode with the `APACHE_MPM="prefork"` directive in the `/etc/sysconfig/apache2` global Apache configuration file.

Apache HTTP Server Configuration

OES configures Apache settings in the `/etc/sysconfig/apache2` global configuration file and the `/etc/apache2/conf.d/oes_httpd.conf` configuration file.

The `/etc/sysconfig/apache2` configuration file controls some global settings of Apache, such as modules to load, additional configuration files to include, server flags to apply when the Apache HTTP Server daemon (`httpd2`) is started, and flags that should be added to the command line.

Apache User wwwrun and Group www

Apache uses the user `wwwrun` identity to serve files to clients of your Web site. OES and Apache configure the following during the OES installation:

- The Apache installation creates a local group `www` and user `wwwrun` on the server.

  You configure the user `wwwrun` as the file owner of the Web site's main directory and files.
- OES creates the group `www` and the user `wwwrun` in Novell eDirectory when you install an OES server in an eDirectory tree for the first time. The user `wwwrun` is added as a member of the group `www`. The user `novlxsrvd` is also created and added to the group `www`.
- OES enables the group `www` and its member users (`wwwrun` and `novlxsrvd`) for Linux with Linux User Management (LUM).

  If your Web site is hosted on an NSS volume or an NCP-enabled Linux volume, you must assign the eDirectory user `wwwrun` as a file system trustee of the Web site's main directory, and give the trustee Read and File Scan rights.

For information about changing the file owner or configuring a file system trustee, see Section 3.2.7, "Configuring Permissions for the Web Site DocumentRoot Directory," on page 28.

Virtual Host for the OES Welcome Web Site

OES automatically configures the OES Welcome Web site in the `/etc/opt/novell/httpd/conf.d/welcome-apache.conf` file. Listening is set up on port 80 in the `/etc/apache2/listen.conf` file. Port 80 is opened in the firewall. The Apache HTTP Server daemon (`httpd2`) starts automatically on server restart.

Apache serves the Welcome page for the OES server at

http://<*server_dns_or_ip_address*>

Secure SSL Virtual Host for the Default Web Site

OES automatically configures a default secure virtual host (`_default_:443`) in the `/etc/apache2/vhost.d/vhost-ssl.conf` file. It sets up listening on port 443 in the `/etc/apache2/listen.conf` file. It opens port 443 in the firewall. The default virtual host configuration is automatically loaded first. It is also used when a domain name does not match a virtual host configuration. The default virtual host defines a custom log `/var/log/apache2/ssl_request_log` to capture events for SSL requests. An `Include` directive in the `/etc/apache2/vhost.d/vhost-ssl.conf` file automatically loads the virtual hosts that are defined in the `/etc/opt/novell/httpd/sslconf.d/*.conf` files.

Secure SSL Virtual Host for the Novell iManager Web Site

If you install Novell iManager on an OES server, the iManager installation automatically configures a secure virtual host for iManager and Novell Portal Services (NPS) in the `/etc/opt/novell/iManager/nps-Apache.conf` file. A symbolic link in the `/etc/opt/novell/httpd/sslconf.d/` directory points to the `nps-Apache.conf` file. This allows the virtual host to be automatically included along with the default secure virtual host when Apache is restarted.

Aliases are defined in the `nps-Apache.conf` file to hit the Web site with any of the following URLs:

https://<*server_dns_or_ip_address*>/nps/iManager.html

https://<*server_dns_or_ip_address*>/nps

https://<*server_dns_or_ip_address*>/iManager.html

3.2.2 Manually Configuring Apache

On OES servers and Novell Open Workgroup Suite (NOWS) Small Business Edition (SBE) servers, you must manually configure Apache settings, OES virtual hosts, and virtual hosts for your personalized Web sites. Use a text editor to create or modify the configuration files, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes.

WARNING: Do not use the *HTTP Server* option in YaST to manage Apache or the virtual host settings on an OES server. It overwrites essential OES settings for Apache and breaks the existing setup. For recovery information, see Section 3.3.1, "Apache Server Errors after Using the HTTP Server Option in YaST," on page 33.

For information about using the configuration files to manage your Apache HTTP Server and virtual hosts, see "Configuring Apache Manually" (https://www.suse.com/documentation/sles11/book_sle_admin/data/sec_apache2_configuration.html#sec_apache2_configuration_manually) in the *SLES 11 Administration Guide* (http://www.suse.com/documentation/sles11/book_sle_admin/data/book_sle_admin_pre.html).

3.2.3 Creating and Configuring a Virtual Host for Each Web Site

On Linux, the Apache HTTP server can serve multiple universal resource identifiers (URIs) from a single instance of Apache running on the server. That is, multiple Web sites, such as www.example.com and www.example.net, can be run from a single Web server. Each Web site is referred to as a *virtual host*. Virtual hosts can be name based, IP based, or port based.

You can set up personalized Web sites by manually creating a virtual host configuration file for each Web site. Templates for secure SSL virtual host and non-secure virtual host configuration files are available in the `/etc/apache2/vhosts.d/` directory.

When you cluster-enable the Web content by using Novell Cluster Services, use the IP address of the cluster resource for the virtual host. This ensures that the Web site traffic is directed to the cluster node where the Web content cluster resource is currently active. Do not use the server node's IP address or the master IP address of the cluster. Specify the Linux file path to the Web content. Linux paths are case-sensitive.

On OES servers, you create and configure a separate virtual host configuration file for each Web site that you want to host in the cluster. The following procedure provides basic information about setting up the file. Refer to other sections in this document to learn about the key settings that are available. For detailed information, see the Apache Virtual Host documentation Web site (http://httpd.apache.org/docs/2.2/vhosts/).

IMPORTANT: The following procedure assumes that the Web site contents reside on a clustered NSS volume. If you use a clustered Linux volume or a clustered NCP-enabled Linux volume, modify the paths according to your configuration.

---

1. Choose an OES node in the cluster, then log in as the `root` user.
2. Create a copy of the virtual host template file in the `/etc/apache2/vhosts.d/` directory.

   The `/etc/apache2/vhosts.d/` directory contains a basic template (`vhost.template`) for a non-secure virtual host and an SSL template (`vhost-ssl.template`) for a secure virtual host.
3. Rename the file with a name for your virtual host, and add the `.conf` file extension, such as `mysite-Apache.conf`.
4. Open the virtual host file in a text editor and configure the virtual host settings for your personalized Web site:

   4a If the Web content is clustered with Novell Cluster Services, set the `VirtualHost` directive to the IP address or DNS host name assigned to the cluster resource:

   ```
   <VirtualHost hostname>
   ```

   For example, if the DNS name is `mysite.example.com`, specify `mysite` as the `VirtualHost`.

   ```
   <VirtualHost mysite>
   ```

   4b Set the value of the `DocumentRoot` directive to the Linux path of the directory where you placed your Web content, and specify the directory options for this location.

   The target directory must contain an `index.html` file, which is the root document for the virtual host. Specify the Linux path to the directory. For example, if you place your Web content in an NSS volume path `APACHEVOL:\www\mysite`, the Linux path is `/media/nss/APACHEVOL/www/mysite`.

   ```
   DocumentRoot "/media/nss/APACHEVOL/www/mysite"

   <Directory  "/media/nss/APACHEVOL/www/mysite">
      # Possible options are "None", "All" or any combination of:
      # Indexes Includes FollowSymLinkx SymLinksifOwnerMatch ExecCGI MultiViews

      Options Indexes MultiViews
      AllowOverride None
      Order allow,deny
      Allow from all
   </Directory>
   ```

4c Configure the host settings as desired for other directives in the file.

The minimum settings for a non-secure Web site are shown in the following example:

```
<VirtualHost mysite>

DocumentRoot "/media/nss/APACHEVOL/www/mysite"

ServerAdmin  mysite-admin@example.com
ServerName   mysite.example.com

ErrorLog     /var/log/apache2/error_log
TransferLog  /var/log/apache2/access_log
#CustomLog   /var/log/apache2/mysite.example.com-access_log combined


HostnameLookups On

UseCanonicalName On

ServerSignature Off

<Directory  "/media/nss/APACHEVOL/www/mysite">
   # Possible options are "None", "All" or any combination of:
   # Indexes Includes FollowSymLinkx SymLinksifOwnerMatch ExecCGI MultiViews

   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

</VirtualHost>
```

4d (Optional) Specify alias paths in the virtual host configuration file.

For example, specify an alias for a Support Web location that has a `support` directory at the same level as `mysite`. Include the `Alias` and `Directory` directives before the `</VirtualHost>` close tag.

```
Alias /support "/media/nss/APACHEVOL/www/support"
<Directory "media/nss/APACHVOL/www/support">
   Options Indexes MultiViews
   AllowOverride None
   Order deny,allow
   Allow from all
</Directory>
```

For information about alias paths that require LDAP authentication, see Section 3.2.8, “Configuring a Web Location that Requires LDAP Authentication,” on page 30.

4e Save the virtual host configuration file.

5 (Optional) In the `/etc/apache2/listen.conf` file, add a `Listen` directive that specifies the IP address that you assigned to your cluster-enabled pool, and specify the port to use.

OES configures Apache to listen on non-secure port 80 by default. It listens for all traffic.

6 Make the Web sites visible on your network or to the world:

6a Add the site name and IP address resolution to your DNS server to make them visible.

6b If you use a non-standard port, open the port in the node’s firewall.

6c If the traffic is from outside the firewall, open the port in the network firewall.

7 Gracefully restart the Apache HTTP Server daemon to apply the virtual host configuration:

```
rcapache2 graceful
```

Each `.conf` file is automatically included in the Apache configuration when you restart Apache.

8 Set up the virtual host for each of the remaining nodes:

8a Log in to the next node as the `root` user.

8b Copy the virtual host configuration file (such as `/etc/apache2/vhosts.d/mysite-apache.conf`) to the next node.

8c Create a local Linux path to the Web site that you specified in the `DocumentRoot` directive and to any paths you specified in `Alias` directives, then make the user `wwwrun` the owner of the directory and its contents.

When Apache is started or restarted, it looks for the paths specified in your Web site's virtual host configuration file. If a path does not exist, Apache reports an error but it loads the virtual host. Users access the site via the IP address or DNS name of the cluster resource, so Web content is served only on the node where the resource is active.

When a cluster resource is not active on a node, the volume subdirectory (such as `APACHEVOL`) in the `/media/nss` directory is normally removed, and the path to the Web site does not exist. Creating the local path allows Apache to find the path even when the resource is not active on the node, and no error is reported when Apache loads. When the resource is taken offline, NSS does not remove the volume directory because it is now non-empty (it contains the local paths you create). The local path should not contain files. To add or remove Web content files, access the NSS volume via the IP address of the cluster resource.

Enter the following commands for the Web site path and alias paths. The `chown` command changes the group to the Apache `www` group unless the group is the `root` user.

```
mkdir -p /media/nss/<volume_name>/<path>

chown wwwrun:www /media/nss/<volume_name>/<path>
```

For example, enter

```
mkdir -p /media/nss/APACHEVOL/www/mysite

chown wwwrun:www /media/nss/APACHEVOL/www/mysite

mkdir -p /media/nss/APACHEVOL/www/support

chown wwwrun:www /media/nss/APACHEVOL/www/support
```

8d Open a terminal console as the `root` user, then gracefully restart Apache:

```
rcapache2 graceful
```

8e Repeat these steps on each of the remaining nodes in turn.

IMPORTANT: Any time that you make changes to the virtual host configuration file, you must copy the modified file to every node in the cluster, and gracefully restart Apache on each node.

3.2.4 Requiring Strong Ciphers

We recommend that you secure your Web solution by requiring strong ciphers when the client is negotiating the connection in the SSL handshake.

We recommend that you enable only the strongest ciphers: RSA, HIGH, and SSLv2.

To enable strong ciphers and disable weak ciphers:

1. In a text editor, modify the `/etc/apache2/vhosts.d/vhost-ssl.conf` file to require strong ciphers. Modify the default settings by placing a plus sign (+) before RSA, HIGH, and SSLv2, and placing an exclamation mark (!) before the weaker ciphers:

   ```
   # SSL Cipher Suite:
   SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:!MEDIUM:!LOW:+SSLv2:!EXP:!eNULL
   ```

2. Gracefully restart Apache on the server:

   ```
   rcapache2 graceful
   ```

3. Repeat this process on every Linux node in the cluster.

   You can alternatively copy the Apache SSL configuration file (`/etc/apache2/vhosts.d/vhost-ssl.conf`) to every Linux node in the cluster, and then restart Apache.

3.2.5 Configuring an SSL Certificate for the Server

OES automatically configures secure SSL communications for a default virtual host (`_default_:443`). SSL is enabled in the Apache global configuration file (`/etc/sysconfig/apache2`) with the following directive:

```
APACHE_SERVER_FLAGS="SSL"
```

The default SSL configuration is defined in the `/etc/apache2/vhosts.d/vhost-ssl.conf` file. It uses an `INCLUDE` directive for the `/etc/opt/novell/httpd/sslconf.d/*.conf` files. This target directory contains the configuration files (or symbolic links to them) for OES virtual hosts that require SSL, such as the `nps-Apache.conf` file that is used for the Novell iManager tool.

By default, OES sets up an SSL certificate file and key file for the server by using certificates generated in Novell eDirectory. Table 3-1 identifies the location of the SSL certificate and key files that are referenced by the `SSLCertificateFile` and `SSLCertificateKeyFile` directives in the `/etc/apache2/vhosts.d/vhost-ssl.conf` file.

Table 3-1 *Default OES Server Certificates*

OES Server Certificate File	Location
SSL Certificate File	`/etc/ssl/servercerts/servercert.pem`
SSL Certificate Key File	`/etc/ssl/servercerts/serverkey.pem`

IMPORTANT: If you use SSL, set up a server certificate for each virtual host unless you use a wildcard certificate.

If you modify the content or location of the certificate and key files, gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the new values.

3.2.6 Configuring Apache to Listen on Multiple Ports

The `Listen` directive in the `/etc/apache2/listen.conf` file tells the Apache HTTP Server to accept incoming requests on the specified port or an address-and-port combination. If the directive specifies only a port, the server listens to the given port on all interfaces. If the directive specifies an IP address and port combination, the server listens on the given port and network interface.

By default, OES configures Apache to listen on non-secure port 80 and secure port 443 in the `/etc/apache2/listen.conf` file. If a firewall is used on the server, port 80 and port 443 are automatically opened in the firewall. The ports are not bound to a particular IP address, so Apache responds to requests on all IP interfaces on the server.

```
Listen 80


<IfDefine SSL>
    <IfDefine !NOSSL>
  <IfModule mod_ssl.c>

      Listen 443

  </IfModule>
    </IfDefine>
</IfDefine>
```

You can configure multiple `Listen` directives to specify multiple IP addresses and ports. The server responds to requests from any of the listed addresses and ports. For information about formats and options for the `Listen` directive, see the *Listen Directive* (http://httpd.apache.org/docs/2.2/mod/mpm_common.html#listen) in the *Apache MPM Common Directives* collection.

If you configure non-standard ports for your personalized Web sites, you must add a `Listen` directive in the `/etc/apache2/listen.conf` file, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes. Ensure that you open the port in the firewall.

3.2.7 Configuring Permissions for the Web Site DocumentRoot Directory

Apache uses the user `wwwrun` identity to serve files to clients of your Web site. You must configure permissions for the Web site content that allow Apache to serve the files to client users.

- "Setting the User wwwrun as the Owner of the Web Site's Directory and Files" on page 28
- "Setting User wwwrun as a File System Trustee of the Web Site's Directory" on page 29

Setting the User wwwrun as the Owner of the Web Site's Directory and Files

The user `wwwrun` must be the file owner of the Web site's main directory and files.

1. Log in as the `root` user, and open a terminal console.
2. Change directory to go to the directory that contains the main directory of your Web site. This is the directory you specify as the `DocumentRoot` in the virtual host configuration file.

   For example, if the `DocumentRoot` is `/media/nss/APACHEVOL/www/mysite`, enter

   ```
   cd /media/nss/APACHEVOL/www
   ```
3. Change the owner of the Web site's directory and files to user `wwwrun`. Enter:

   ```
   chown -R wwwrun:www mysite
   ```

This recursively modifies the owner to user `wwwrun` for the directory and the subdirectories and files it contains. It changes the group to `www` unless the group is set to the `root` user.

4 In a file browser, view the directory's properties to verify that the owner was changed.

You can also use the `ls -al <path>` command to list the directory and view the owner, group, and permissions.

Setting User wwwrun as a File System Trustee of the Web Site's Directory

OES automatically creates the user `wwwrun` and group `www` in Novell eDirectory. Both are LUM-enabled. You can verify their configuration by using the *Directory Administration* option and *Linux User Management* option in Novell iManager.

If your Web site is hosted on an NSS volume or an NCP-enabled Linux volume, you must assign the eDirectory user `wwwrun` as a file system trustee of the Web site's main directory, and give the trustee Read and File Scan rights. You can also set the `www` group as a trustee with Read and File Scan rights.

1 Log in to Novell iManager as an administrator user.

2 In the iManager toolbar, click the *View Objects* icon.

3 In the Tree view, select the volume, then browse the file system to locate the directory that contains your Web site's content.

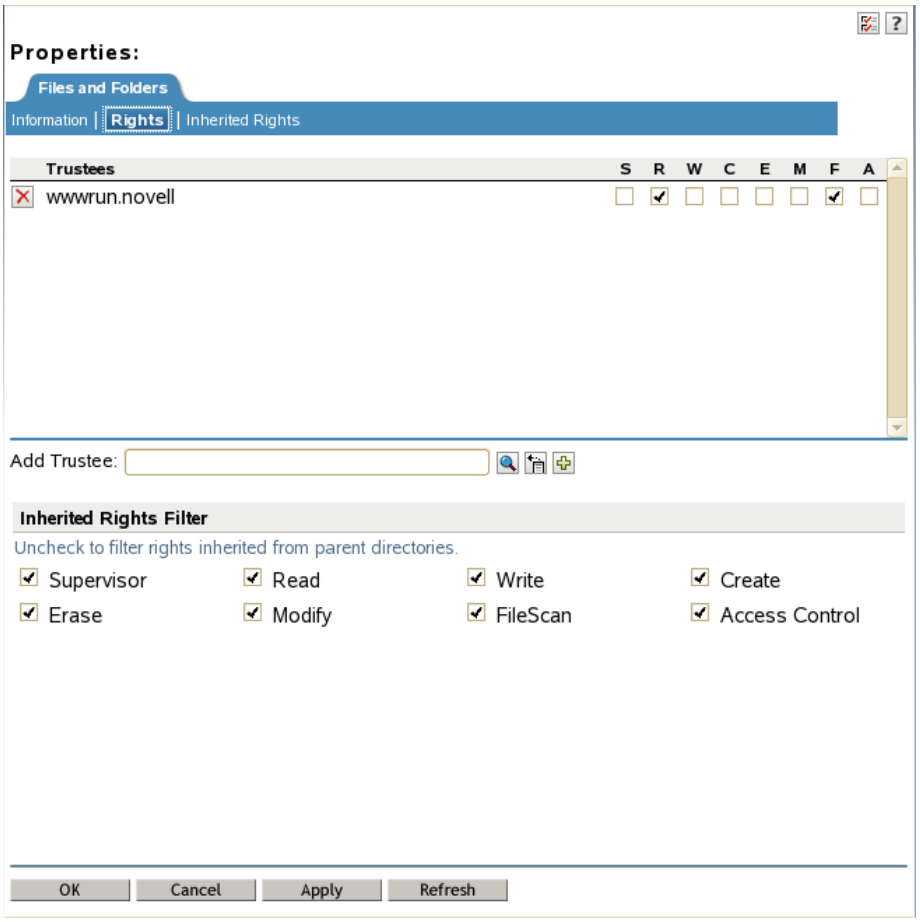
4 Select the check box next to the directory, then select *Actions* > *Properties*.

5 On the Properties page, select *Rights*.

6 Click the *Add Trustee* browse icon to open the *Object Selector*.

7 Locate and select the user `wwwrun`, then click *OK*.

The user `wwwrun` is added as a trustee with the default Read and File Scan rights.



8 Click *Apply* or *OK* to save the changes.

3.2.8 Configuring a Web Location that Requires LDAP Authentication

If you have documents or a location that requires restricted Web access, you can set up Apache to enforce eDirectory authentication and force the authentication to be done over `https`. This solution can be used on individual directories, URLs, or the entire Apache server.

The following example creates a single secure location so that any document that is referenced under the directory requires authentication. For example, the URL www.example.com can have public access, while the URL www.example.com/secure and documents it contains require authentication. Authentication should be done over a secure connection (https) rather than a non-secure connection (`http`). All `http` attempts are redirected to `https` for the given location.

1 Ensure that the `rewrite` module is enabled in the `/etc/sysconfig/apache2` global configuration file. OES enables this module by default.

   Open the `/etc/sysconfig/apache2` file in a text editor, and verify that `rewrite` is listed in the modules defined in the `APACHE_MODULES` directive.

2 Configure the permissions for the user `wwwrun` on the target directory:

   2a Change the owner to the Apache user `wwwrun`:

```
chown -R wwwrun:www /media/nss/APACHEVOL/www/secure
```

This changes the group to the Apache group `www` unless the group is the `root` user.

2b For an NSS volume or an NCP-enabled Linux volume, configure the user `wwwrun` as a file system trustee of the `/media/nss/APACHEVOL/www/secure` directory, and give the trustee Read and File Scan rights.

For information, see "Setting User wwwrun as a File System Trustee of the Web Site's Directory" on page 29.

3 In a text editor, create a copy of the `/etc/apache2/vhosts.d/vhosts-ssl.template` file to create a `secure.conf` configuration file.

4 Allow for all `http` requests for the `/secure` alias to be redirected to `https`. Add the following directives to the `secure.conf` file:

```
RewriteEngine On

RewriteRule ^/secure https://%{SERVER_NAME}/secure [L,R]
```

5 If the location that contains secure information exists outside the `DocumentRoot` directory, create an alias to the directory. Add the following line to the `secure.conf` file:

```
Alias /secure "/<path_to_directory>/secure"
```

For a cluster resource, the secure directory ideally resides on the same clustered volume as the Web site, and at the same directory level as `DocumentRoot` for the Web site:

```
Alias /secure "/media/nss/APACHEVOL/www/secure"
```

6 Under the `Alias` directive, add the option for LDAP authentication under the `Directory` directive in the `secure.conf` file. Specify the IP address or DNS name of the Web site's cluster resource.

```
<Directory "media/nss/APACHVOL/www/secure">
  Options Indexes MultiViews
  AllowOverride None
  Order deny,allow
  Allow from all
  AuthType Basic
  AuthName "Protected"
  require valid-user
  AuthLDAPAuthoritative On
  AuthLDAPURL ldaps://<cluster_resource_ip_address_or_dns_name>/o=corp?uid?sub
</directory>
```

7 Save the `/etc/apache/vhosts.d/secure.conf` file.

8 Open a terminal console as the `root` user, then gracefully restart the Apache daemon:

```
rcapache2 graceful
```

9 Verify that Apache is able to start.

If there are errors, make corrections in the configuration file, then restart the Apache daemon.

10 In a Web browser, go to the Web site with `http` and verify that you are redirected to `https`, and that you can authenticate against the `/secure` alias.

3.2.9 Starting, Stopping, or Restarting the Apache Daemon

The Apache HTTP Server program runs as a daemon (`httpd2`) that executes continuously in the background to handle requests. OES configures the daemon to start automatically on system restart. You must restart Apache to apply any changes you make to the Apache or virtual host configuration files, or to add new virtual host configuration files. A graceful restart does not disrupt the service.

In a cluster, you manually copy the virtual host configuration files for clustered personalized Web sites to every node in the cluster. When Apache starts on each node, it reads the configuration file and is available to serve the site when the resource is active on the node. You do not add Apache commands in the resource's load and unload scripts. All requests to a clustered Web site are sent to the DNS name or IP address of the cluster resource, and not to a specific node. The site's requests are served by the Apache process that runs on the node where the cluster resource is currently active.

To start, stop, or restart the Apache daemon, use the `/usr/sbin/rcapache2` commands in Table 3-2:

Table 3-2 */usr/sbin Commands*

Command	Description
`rcapache2 start`	Starts the `httpd2` parent process. The parent process reads its configuration files and opens log files, and then spawns the child processes to serve hits. OES configures the Apache daemon to start automatically on server restart.
`rcapache2 stop`	Causes the parent process to immediately attempt to kill all of its child processes. This can take several seconds. The parent exits after all child processes have exited. Any requests in progress are terminated, and no further requests are served.
`rcapache2 graceful-stop`	Causes the parent process to advise its child processes to exit after their current request (or to exit immediately if they are not serving anything). The parent removes its PID file and ceases listening on all ports. The parent continues to run, and monitors child processes that are handling requests. The parent exits after the child processes complete the pending requests and exit, or when a timeout period has elapsed (as specified by the `GracefulShutdownTimeout`). If the timeout is reached, any remaining child processes are automatically sent the TERM signal to force them to exit, and any requests in progress are terminated.
`rcapache2 restart`	Causes the parent process to immediately kill its child processes like the `stop` option, but the parent does not exit. It re-reads its configuration files, and re-opens any log files. Then it spawns a new set of child processes and continues serving hits.
`rcapache2 graceful`	Causes the parent process to advise the child processes to exit after their current request (or to exit immediately if they are not serving anything). The parent re-reads its configuration files and re-opens its log files. As each child dies, the parent replaces it with a child from the new generation of the configuration, which begins serving new requests immediately.

3.2.10 Viewing the Apache Log Files

The following Apache log files are located in the `/var/log/apache2/` directory:

```
access_log
```

```
error_log
rcapache2.out
rewrite_log
ssl_request_log
```

You can also specify custom logs by adding the `CustomLog` directive to your virtual host configuration file. For information about formatting the custom log, see *Apache Module mod_log_config* (http://httpd.apache.org/docs/2.2/mod/mod_log_config.html).

3.3 Troubleshooting the Apache HTTP Server

This section describes some issues you might experience with Apache HTTP Server and provides suggestions for resolving or avoiding them. For additional troubleshooting information, see the Novell Technical Support Knowledgebase (http://www.novell.com/support).

3.3.1 Apache Server Errors after Using the HTTP Server Option in YaST

If you use the *HTTP Server* option in YaST to manage Apache or virtual hosts, the option can overwrite essential OES settings and load the wrong modules, which breaks the default Apache HTTP Server setup. For information, see TID 7002562 (http://www.novell.com/support/kb/doc.php?id=7002562) in the Novell Knowledgebase.

If you have used the *HTTP Server* option in YaST and Apache is no longer working, recover the OES default Apache HTTP Server setup by doing the following:

1. As the `root` user, open the `/etc/sysconfig/apache2` file in a text editor and modify the following directives:
   - **Proxy module:** In the `APACHE_MODULES=` line in the file, ensure that the `proxy` module is listed before the `proxy_ajp` module. For example (some modules are not listed for ease of reading the example):

     ```
     APACHE_MODULES="cgi dir rewrite ssl proxy proxy_ajp ssl"
     ```

   - **SSL module:** In the `APACHE_MODULES=` line in the file, ensure that the `ssl` module is listed. For example (some modules are not listed for ease of reading the example):

     ```
     APACHE_MODULES="cgi dir rewrite ssl proxy proxy_ajp ssl"
     ```

   - **Prefork mode:** If OES installs Apache Prefork packages, Apache should run in `prefork` mode rather than `worker` mode. To force this, ensure that the `APACHE_MPM=""` line is set to `"prefork"`. For example:

     ```
     APACHE_MPM="prefork"
     ```

   - **SSL:** Ensure secure communications by enabling the SSL flag. For example:

     ```
     APACHE_SERVER_FLAGS="SSL"
     ```

2. Gracefully restart Apache to apply the changes. As the `root` user, enter the following command at a console prompt:

   ```
   rcapache2 graceful
   ```

3.3.2 Files Downloaded from NetStorage Are 0 Bytes

After you lock down ciphers for an Apache HTTP Server to use only the strongest SSL ciphers, all of the files downloaded from NetStorage are 0 bytes in size.

NetStorage might not work as expected if you lock down Apache HTTP Server to disallow low and medium SSL ciphers. Try allowing medium SSL cipher settings to see if that is sufficient, then add back low cipher settings if necessary.

For other SSL cipher configuration options, see *SSL/TLS Strong Encryption: How-To* (http://httpd.apache.org/docs/2.2/ssl/ssl_howto.html) at Apache.org.

3.4 Additional Information

The latest Apache documentation is available on the Apache HTTP Server Version 2.2 Documentation Web site (http://httpd.apache.org/docs-2.2/).

A Documentation Updates

This section contains information about documentation content changes made since the initial release of Novell Open Enterprise Server. If you are an existing user, review the change entries to readily identify modified content. If you are a new user, simply read the guide in its current state.

This document was updated on the following dates:

A.1 May 3, 2003

Section 2.1, "What's New (May 2013)," on page 17 is new.

A.2 April 12, 2013

Chapter 3, "Configuring Apache HTTP Server on OES Servers with Novell Cluster Services," on page 19 is new.

A.3 June 21, 2012

We recommend that you use strong ciphers for your Apache Web Server. See Section 3.2.4, "Requiring Strong Ciphers," on page 26.

A.4 November 21, 2011

In addition to bug fixes, Novell Cluster Services added support for OES 2 SP3 services and file systems on the SUSE Linux Enterprise Server (SLES) 10 SP4 operating system. You can upgrade to SLES 10 SP4 by using the move-to-sles10-sp4 patch in the SLES patch channel.

Links have been altered to the SLES 10 SP4 documentation Web site.

A.5 December 2010 (OES 2 SP3)

This guide was updated to conform with Novell documentation standards. Information specific to the NetWare 6.5 SP8 operating system was removed. For NetWare Web information, see the *NW 6.5 SP8: Web and Application Services Overview*.

A.6 November 9, 2009 (OES 2 SP2)

Updated to the revised documentation standards.

A.7 December 2008 (OES 2 SP1)

Updated to the revised documentation standards

A.8 December 1, 2005 (OES 2)

Page design reformatted to comply with revised Novell documentation standards.