Novell Web Search Server

3.0

www.novell.com

ADMINISTRATION GUIDE

103-000144-001

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.

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

U.S. Patent No. 5,157,663; 5,349,642; 5,455,932; 5,553,139; 5,553,143; 5,572,528; 5,594,863; 5,608,903; 5,633,931; 5,652,854; 5,671,414; 5,677,851; 5,692,129; 5,701,459; 5,717,912; 5,758,069; 5,758,344; 5,781,724; 5,781,733; 5,784,560; 5,787,439; 5,818,936; 5,828,882; 5,832,274; 5,832,275; 5,832,483; 5,832,487; 5,859,978; 5,870,561; 5,870,739; 5,873,079; 5,878,415; 5,884,304; 5,893,118; 5,903,650; 5,903,720; 5,905,860; 5,910,803; 5,913,025; 5,913,209; 5,915,253; 5,925,108; 5,933,503; 5,933,826; 5,946,002; 5,946,467; 5,956,718; 5,956,745; 5,964,872; 5,974,474; 5,983,223; 5,983,234; 5,987,471; 5,991,810; 6,002,398; 6,014,667; 6,016,499; 6,023,586; 6,029,247; 6,052,724; 6,061,726; 6,061,740; 6,061,743; 6,065,017; 6,081,774; 6,081,814; 6,094,672; 6,098,090; 6,105,062; 6,105,069; 6,105,132; 6,115,039; 6,119,122; 6,144,959; 6,151,688; 6,157,925; 6,167,393; 6,173,289; 6,216,123; 6,219,652; 6,233,859; 6,247,149; 6,269,391; 6,286,010; 6,308,181; 6,314,520; 6,324,670; 6,338,112; 6,345,266; 6,353,898; 6,424,976; 6,466,944; 6,477,583; 6,477,648; 6,484,186; 6,496,865; 6,510,450; 6,516,325; 6,519,610; 6,532,451; 6,532,491; 6,539,381; RE37,178. Patents Pending.

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

www.novell.com

NetWare Web Search Server Administration Guide

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

Third-Party Trademarks

About This Guide

This guide describes how to use NetWare Web Search Server to add search functionality to your Internet or Intranet Web site. It is intended for use by anyone involved in installing, managing, and using NetWare Web Search Server to create search services. It is divided into the following sections:

- Chapter 1, “Overview,” on page 7
- Chapter 3, “Installation and Configuration,” on page 15
- Chapter 4, “Designing Your Search Solution,” on page 23
- Chapter 5, “Creating and Managing Virtual Search Servers,” on page 29
- Chapter 7, “Optimizing Search Results,” on page 49
- Chapter 6, “Understanding Templates,” on page 43
- Chapter 9, “Customizing Your Search Solutions,” on page 77
- Chapter 8, “Working with Template Variables and Search Parameters,” on page 59
- Chapter 10, “Internationalizing Your Search Solution,” on page 81
- Appendix A, “Troubleshooting NetWare Web Search,” on page 89
- Appendix B, “Combined Character Sets for Use with NetWare Web Search,” on page 91

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

Also, a trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

1 Overview

NetWare® Web Search 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.

This section describes what's new in this release, how Web Search works, and how you can use it to build customized search services for use by your own company and its customers.

Benefits of NetWare Web Search Server

NetWare® Web Search Server offers a powerful full-text search engine you can use to add search capabilities to your Internet or intranet Web sites. Compatible with the Apache Web server, included with NetWare 6.5, you can create custom search forms and search result pages either from scratch or by using the templates provided with NetWare Web Search Server.

With NetWare Web Search Server, you can

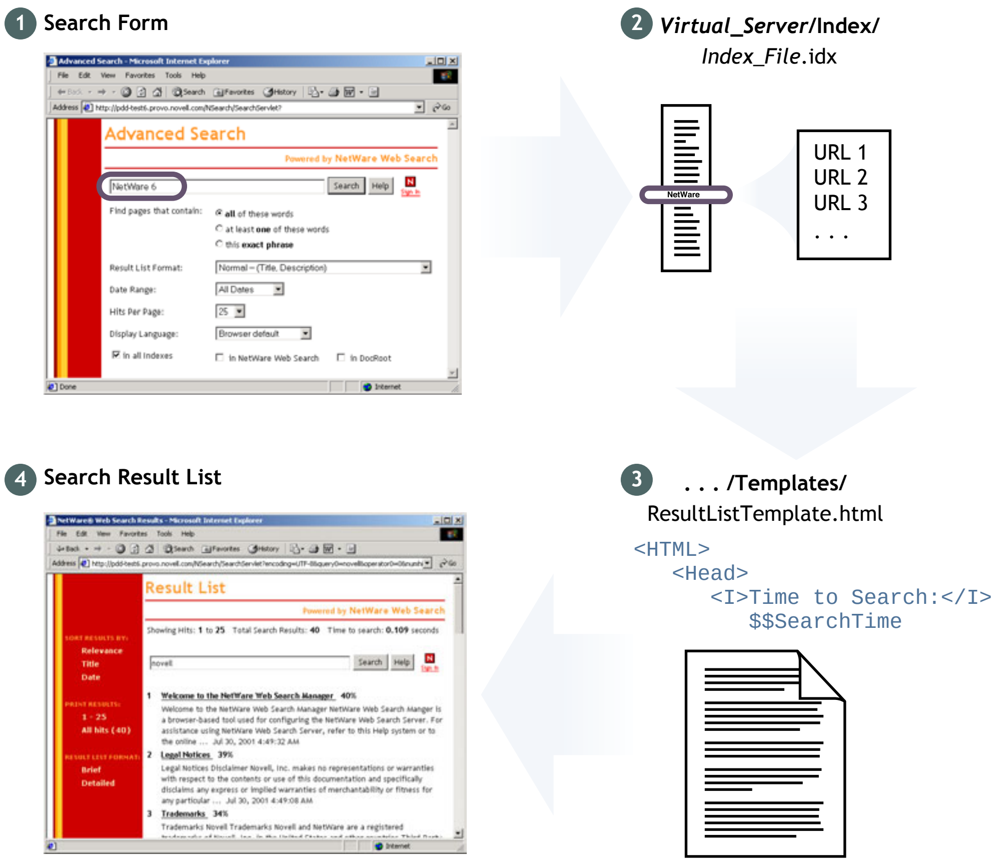
- Search across multiple Web sites, servers, and file formats in any language, all from a single interface.
- Host search services for one or more companies or organizations.
- Print large collections of dispersed but related files as a single, coherently organized document.
- Customize the look and feel of search and print results in all languages.
- Create themes, which are defined collections of search and print result templates that allow you to deploy custom look and feel virtual search servers, each for a specific company or department.

How NetWare Web Search Works

Understanding how Web Search handles searches can help explain the role of templates, template variables, and query parameters in Web Search. One of the great benefits of Web Search is the simplicity of customizing it to meet your needs or the needs of your clients.

The following figure shows what happens when users submit words through the search template and how Web Search then handles the words to generate and display search results.

Figure 1 How NetWare Web Search Handles a Search String



```
<HTML>
   <Head>
      <I>Time to Search:</I>
         $$SearchTime
```

In this diagram, the user (1) enters a search string such as `NetWare 6.5`. The search string is then searched for in the index file on the Web Search Server (2). If the search string is located, the Uniform Resource Locators (URLs) and document titles and descriptions are passed on to the search results template (3) and displayed to the user (4). The information that is displayed is determined by the variables included in the search results template, which means that you can modify what information is actually returned to the user by adding or removing Web Search variables from the template.

Components of a Virtual Search Server

Similar in purpose to a software virtual server, a *virtual search server* is a way to group similar information together for a specific purpose and audience. For example, you might create one virtual search server for your company's support organization, another for it's public Web site, and yet another for your intranet. You might break these down even further by creating more focused virtual search servers for groups within these organizations.

Literally, a virtual search server is a collection of HTML templates and supporting data files and consists of the following components:

- A name and (optional) alias for accessing a virtual search server
- Index files containing key words and related URLs for use in search results
- Scheduled indexing events
- Search and print results templates

- Log files

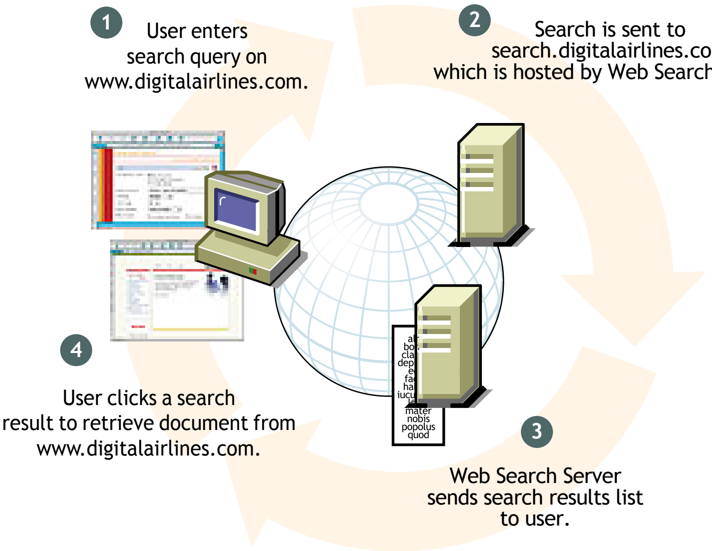
Each of these components are managed through the NetWare Web Search Manager, which is accessed using a Web browser (see Chapter 5, “Creating and Managing Virtual Search Servers,” on page 29).

HINT: If you want two or more virtual search servers to share an index, create a duplicate index on each virtual search server that points to the same index directory. In this manner, all virtual search servers can search a shared index in addition to their own indexes.

General Architecture of a Web Search Service

The general architecture of a Web search service is depicted in the following diagram.

Figure 2 Search Service Architecture


1
User enters search query on www.digitalairlines.com.
2
Search is sent to search.digitalairlines.com which is hosted by Web Search Server.
3
Web Search Server sends search results list to user.
4
User clicks a search result to retrieve document from www.digitalairlines.com.

In Figure 2, the user enters a search query in a search form found on www.digitalairlines.com (1 and 2). When the user clicks the Search button, the query is sent to search.digitalairlines.com, which is hosted by the NetWare Web Search Server (3), which then processes the query using index files that were created using the Web Search Manager.

Web Search then compiles the results of the search into an HTML template, which has been modified to match the look and feel of www.digitalairlines.com, and returns them to the user’s Web browser (4).

When users click a search result link to view the content, they are taken to that content, whether it is hosted on www.digitalairlines.com or some other Web site.

Building a Virtual Search Server

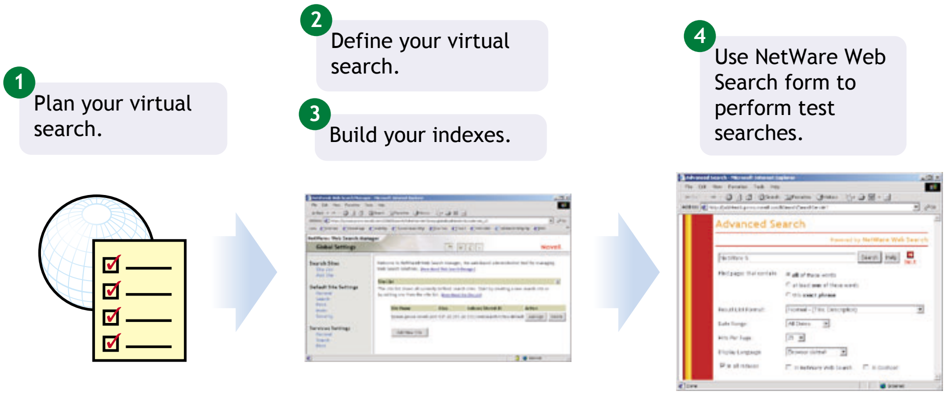
Using NetWare Web Search Manager, you define a new virtual search server and then index data found on your NetWare server or content found on any Web site that can then be searched using the NetWare Web Search Server.

Building a virtual search server involves four fundamental tasks:

1. Plan the purpose of your virtual search server, which includes identifying what will be searched and how to optimize your search solution based on the content to be searched.
2. Defining a new virtual search server using Web Search Manager.
3. Building one or more indexes of your Web sites and file servers.
4. Testing your new virtual search server using the default search form and the search and print result templates.

After completing the first task of planning your virtual search server (see Chapter 4, “Designing Your Search Solution,” on page 23), use the NetWare Web Search Manager to complete the remaining tasks. The following figure illustrates these tasks.

Figure 3 Steps to Creating a Virtual Search Server


1
Plan your virtual search.
2
Define your virtual search.
3
Build your indexes.
4
Use NetWare Web Search form to perform test searches.
Advanced Search

Repeat this process for each new virtual search server you want to add. A virtual search server can include one or more indexes of files located on your file server and files located in one or more directories on one or more Web servers.

Indexing a Web server (or Web site) involves a process known as *crawling*. The Web Search Server begins indexing files on a Web server at the directory level you specify and continues to index along hypertext links until reaching a dead-end, which occurs when either a linked file cannot be found or when there are no more links found within the specified Web sites.

Accessing NetWare Web Search Manager

NetWare Web Search Manager is the tool you use to create and manage virtual search servers and their associated indexes.

To run NetWare Web Search Manager, do the following:

1 Type `https://`*`domainname`*`:`*`portnumber`* in your Web browser’s address field, where *portnumber* is the port number of NetWare Web Manager and press Enter.

HINT: You must use the HTTPS protocol because NetWare Web Search Manager uses Secure Sockets Layer (SSL). You can disable encryption from the Admin Preferences page of the Apache Manager. See Managing Apache Web Server Preferences in *Apache Web Server Administration Guide*.

2 Under NetWare Web Search Server, click the *servername* link, where *servername* is the name of your NetWare Web Search Server.

Taking a Test Run

When you install NetWare Web Search Server, some of your server's content is automatically indexed and appears on the default search form as the "NetWare Web Search" and "Doc Root" indexes.

Once you start the Enterprise Web Server, you can open the search page using your Web browser and perform a search against the content that has been automatically indexed.

To test NetWare Web Search using the default search page, do the following:

1. Type **`http://`*`domainname`*`/novellsearch`** in your Web browser's address field and press Enter.

   IMPORTANT: The URL is case sensitive. Use the exact case shown above.

2. Type NetWare in the Search field > press Enter.

HINT:

The Search form template, SearchTemplate.html, is stored on your server at /*searchroot*/TEMPLATES. See "Customizing the Look and Feel of Search Server Content" on page 11 for information about how to customize templates.

Customizing the Look and Feel of Search Server Content

Once you've created one or more virtual search servers and built indexes for them, you can customize them by modifying the default search form and result and print templates, or by creating a new templates from scratch using the variables and parameters described in Chapter 8, "Working with Template Variables and Search Parameters," on page 59.

For more information about modifying the default search form and the search and print templates to create your own custom search solution, see Chapter 9, "Customizing Your Search Solutions," on page 77.

For more information about building professional corporate and hosted search services, see Chapter 4, "Designing Your Search Solution," on page 23.

What's Next

If you have not already installed Web Search Server, see Chapter 3, "Installation and Configuration," on page 15.

Once Web Search Server is installed and running, it takes only minutes to create your own virtual search server. For details, see Chapter 5, "Creating and Managing Virtual Search Servers," on page 29.

To learn more about preliminary steps to setting up an optimized search solution, see Chapter 4, "Designing Your Search Solution," on page 23.

2 What's New

Since NetWare® 6, NetWare Web Search Server has been significantly enhanced with new features that speed up search results, improve the accuracy of search results, and enhance the experience your users have while searching.

The following list highlights key new features and enhancements to NetWare Web Search Server included with NetWare 6.5:

- *Stop Words*: Although invisible to the user, this feature significantly improves search times by ignoring insignificant search words, such as conjunctions. For example, removing the word *to* from a user's query can reduce the average query time from 1.7 seconds to 1.08 (a 40% improvement). A set of common stop words are included, but you can easily add your own, or even remove the one or more of the ones we have included. (See "Using Stop Words Processing" on page 51.)
- *Search Expansion Module*: If a user's search query results in a "Not Found" message, NetWare Web Search automatically performs a second search, broadening the search to include additional indexes not searched the first time. (See "Helping Users Around the Not Found Message" on page 54.)
- *URL Redirection*: Lets you specify key words that are to redirect the user's Web browser to a specific URL. For example, searching for *one net* on Novell.com redirects your browser to a special page designed to emphasize Novell's One Net strategy, rather than returning results on the search results page. ("Redirecting A Search" on page 53.)
- *Search Within Searches*: Lets users perform a second search from within the results of an original search. ("Search Within Search" on page 54.)
- *Search Terms Highlighter*: Highlights the search terms and phrases wherever they appear in the actual documents found during the search without affecting the formatting of the original documents. As an administrator, you can specify characteristics of the highlighting, including the markup color. Search terms found in hidden document summary fields and META tags are also highlighted. (See "Highlighter" on page 54.)
- *Web Search Synchronization*: Lets you designate one Web Search server as the search master from which updated indexes, templates, and configuration settings are systematically sent out to all other Web Search servers defined as part of a Web Search Synchronization cluster. This saves system resources on all other machines because they do not have to regenerate indexes themselves. It also lets an administrator manage all other Web Search servers from a single interface. (See "Synchronizing Data Across Multiple Web Search Servers" on page 24.)
- *Synonyms*: When enabled, this feature returns documents in the search results that contain synonyms of the user's original search terms. Search results might be less relevant when this feature is enabled, but it will also help identify content that might not otherwise be found due to the choice of search terms. While common synonyms are included, you can specify additional synonyms. (See "Using Synonyms To Broaden Search Results" on page 53.)

- *Integration with Novell Portal Services*: Add a Web Search gadget to your portal services solution, allowing you to index and search for information located in Web and files servers. May be part of a federated search that includes a GroupWise message store. (See Configuring Gadgets (http://www.novell.com/documentation/lg/portal/configure/data/adat459.html).
- *Enhanced Access Management*: In NetWare 6, Web Search depended on Enterprise Web Server NLMs to authorize each search result for a particular user for those documents managed by eDirectory™. Web Search Server now depends on the rights engine built into NetWare 6.5. In addition to other enhancements, this allows user privileges to entire indexes rather than to each search result, which improves the over all speed at which search results requiring authentication are returned to the user. (See "Restricting Search Results to Specific Areas" on page 35.)
- *Sneak-Peak*: Lets you see a preview of the actual documents listed on the search results page without opening a separate window or leaving the search results page. (See "Sneak Peek" on page 54.)
- *Better Installation Program*: NetWare Web Search Server is included as a patterned deployment option, letting you specify a NetWare server as a dedicated NetWare Web Search Server, optimized for running NetWare Web Search Server. (See Chapter 3, "Installation and Configuration," on page 15).
- *E-mail Notification*: Notifies administrators through e-mail about issues such as index failure, outdated indexes that are being used, or, if you are using Web Search Synchronization, you can be notified when a new index is being pushed out from the search master to other search servers. (See "E-mail Feature" on page 26.)

What's Next

- For information about installing Web Search, see Chapter 3, "Installation and Configuration," on page 15.
- For information about designing a search solution, see Chapter 4, "Designing Your Search Solution," on page 23.
- For more information about Web Search, visit the Web Search Product Page (http://www.novell.com/products/websearch).

3 Installation and Configuration

You can install Web Search Server as a service on your NetWare 6.5 server, or you can dedicate a server to it using the Single Purpose install option. Once installed, you can modify the global settings of your new Web Search server. Global settings control the default settings of all new virtual search servers you create, or in some cases disable them.

This chapter describes how to install Web Search from the NetWare 6.5 Products CD, and then how to configure global (default) settings.

Installing NetWare Web Search Server

If you did not install NetWare Web Search Server during the NetWare 6.5 installation, you can install it at anytime from the *NetWare 6.5 Products* CD.

For more information about setting up a dedicated Web Search server, see the *NetWare 6.5 Overview and Installation Guide*.

To install NetWare Web Search Server from the *NetWare 6.5 Products* CD:

1. Insert the *NetWare 6.5 Products* CD into the drive of the server where you want to install Apache.
2. Start the NetWare GUI by typing **`startx`** at the system console.
3. Click Novell > Install > Add.
4. In the Source Path dialog box, enter the path to the CD, or click the Browse button and locate the CD.
5. Select the POSTINST.NI response file and click OK.
6. From the products list, select Apache2 Web Server.
7. Click Next.
8. When prompted, enter your administrator username and password, and your user context.
9. Click OK.
10. Follow the remaining screen prompts.

IMPORTANT: After the installation is complete, you must restart the NetWare server. Type **`restart`** at the system console.

For instructions about accessing the Web Search Server Manager, the user interface for configuring and managing Web Search, see "Taking a Test Run" on page 11.

IMPORTANT: If you are upgrading from a previous version of Web Search Server, you must regenerate any older indexes before they can be searched.

Configuring NetWare Web Search Server

Before you start creating virtual search servers and building indexes for them, you might want to modify the global settings of your search server, which affect all virtual search servers you create.

Configuring Default Virtual Search Server Settings

NetWare Web Search Manager's home page displays a list of all virtual search servers that exist on your Web Search server. This home page is called Global Settings because the changes you make from this page affect all new virtual search servers that you create, and they also affect the functionality of the search and print servlets that provide the Web Search services.

For example, if you changed the Default Query Encoding under Default Settings > General, any new virtual search servers you create would default to the new setting.

However, you can override default virtual search server settings by using the Advanced index definition pages when defining indexes for your virtual search servers.

Configuring General Settings

Changes you make to the query, response, and error log settings affect all newly created virtual search servers.

To modify general default virtual search server settings, do the following:

1. From the Web Search Manager home page, click General under Default Settings.
2. From the Default Query Encoding drop-down list, select an encoding that represents the character set encoding that most of your user queries will use.
3. In the Maximum Query Duration field, enter the maximum number of seconds before Web Search should end a query, regardless of whether a search has been completed.

   This option is one of several methods for letting you protect your server's resources from processing potential rogue searches, which are sometimes intended to harm your service by consuming server resources.
4. Under Response Settings, select an output encoding from the Default Encoding for Response Pages drop-down list.

   This setting specifies the encoding Web Search should use when responding to user queries using the search and print results templates, and the error and response messages templates.
5. Enter the maximum number of queries in the Refuse Queries if Potential Hits Exceed field to cancel the processing of search results that might take a long time to complete.
6. In the Maximum Log Size field, enter the maximum size (in bytes) that Web Search will allow the log file to grow to.

   Depending on the number of visitors that your virtual search server hosts, log files can become large. This setting will protect your system's hard drive resources.

Configuring Search Settings

To modify default search settings for search features, do the following:

1. From the Web Search Manager home page, click Search under Default Settings.
2. Under Query Results Settings, enter the number of search results in the Default Number of Results to Display field that you want displayed on each search results page.

For example, if you set this to 25 (which is the default setting) and the number of hits in a return was 200, Web Search would only return 25 hits per search results page at a time.

3 Set a limit on the number of results allowed at one time on the results page by entering a number in the Maximum Number of Results to Display field.

4 Enter the highest number of search results that can be returned to a user query in the Highest Allowed Result Number field.

5 Under Template Settings, enter a path to where your Web Search templates are stored in the Templates Directory field.

HINT: The default path is *volume*:\searchroot\Templates, but if you have created custom templates or for some reason want to keep your templates elsewhere, specify the path here so that Web Search knows where the templates are.

6 From the Default Encoding for Templates drop-down list, select the character set that your templates are written in.

This value will be used even with templates that do not specify an encoding. Encodings found in templates that do not match the encoding you specify here will override this encoding.

7 In the Default Search Page Template field, enter the filename of the search page template you want to use.

If you have created a custom template and want Web Search to use it as your search page, enter its name in this field.

8 In the Default Search Results Template field, enter the filename of the search results template you want to use.

If you have created a custom search results template and want Web Search to use it as your default search results page, enter its name in this field.

9 In the Template to Use If No Results Returned field, enter the filename of the template that Web Search should return if no results are found.

10 In the Template to Use If Error Occurs field, enter the filename of the template that Web Search should return if there are errors while processing a user's query.

11 Click Apply Settings.

Configuring Print Settings

To modify default print settings, do the following:

1 From the Web Search Manager home page, click Print under Default Settings.

2 Under Print Results Settings, enter the number of print results in the Default Number of Results to Print field that you want displayed on each print results page.

For example, if you set this to 25 (which is the default setting) and the number of hits in a return was 200, Web Search would only return 25 hits per print results page at a time.

3 Set a limit on the number of results allowed at one time on the results page by entering a number in the Maximum Number of Results to Print field.

4 Enter the highest number of search results that can be returned to a user query in the Highest Allowed Result Number field.

5 To limit the size of a print job, specify the largest print job size that Web Search will allow in the Maximum Print Job Size field.

Any users requesting a print job larger than this value will receive a message informing them that their request was too large.

HINT: This is a useful feature to administrators who want to keep down the size of print jobs in their own companies, departments, or organizations.

6. To be notified when a print job exceeds a certain size, enter the print job size in the Print Job Size Warning field.

   By default, this message is sent using the ResponseMessageTemplate.html file and is intended as a warning to users that they are exceeding the allowed print job size. It then prompts the user to confirm the print job before continuing.

7. Under Template Settings, enter a path in the Templates Directory field to where your Web Search templates are stored.

   HINT: The default path is *volume*:\searchroot\Templates, but if you have created custom templates or for some reason want to keep your templates elsewhere, specify the path here so that Web Search knows where the templates are.

8. From the Default Encoding for Templates drop-down list, select the character set that your templates are written in.

   This value will be used even with templates that do not specify an encoding. Encodings found in templates that do not match the encoding you specify here will override this encoding.

9. In the Default Print Results Template field, enter the filename of the print results template you want to use.

   If you have created a custom print results template and want Web Search to use it when returning print results, enter its name in this field.

10. In the Template to Use If No Results Returned field, enter the filename of the template that Web Search should return if no print results match a user's query.

11. In the Template to Use If More Information Is Needed field, enter the filename of the template to be sent back to users whose print jobs exceed the size you specify in the Print Job Size field. (See Step 6.)

12. In the Template to Use If Error Occurs field, enter the filename of the template that Web Search should return if there are errors while processing a user's print query.

13. Click Apply Settings.

Configuring Index Settings

These settings are intended to make the process of creating indexes even easier by letting you configure common settings as default settings. This saves you time by not making you make the same selections each time you create a new index.

To modify default index settings, do the following:

1. From the Web Search Manager home page, click Index under Default Settings.

2. Select the type of index that you want as the default index type on the Indexing Management page.

3. Check the URLs Are Case Sensitive check box if you want Web Search to recognize URLs that are different only in character case, but are otherwise identical. For example, www.digitalairlines.com verses www.DigitalAirlines.com.

   IMPORTANT: By setting this option to No can help Web Search to avoid indexing duplicate information, which can come from indexing URLs that are presented using different cases but actually point to the same information. However, if a Web server being indexed is configured to differentiate between cases, Web Search might leave out content that you want indexed.

4 Check the Crawl Dynamic URLs (URLs Containing '?') check box if you want dynamic content indexed, in addition to static content.

See "About Indexing Dynamic Web Content" on page 40.

5 Enter a number (in bytes) in the Maximum File Size to Index field to keep Web Search from indexing files larger than the number you specify.

6 In the Maximum Time to Download a URL field, enter a number (in seconds) before Web Search automatically skips the indexing of the specified URL.

7 From the Encoding (If Not in META Tags) drop-down list, select the encoding to be used when indexing files that do not contain an encoding specification.

For example, HTML files can specify their encoding with a Content-Type META tag.

8 Click Apply Settings.

Configuring Security Settings

Security settings let you manage access to indexed content by requiring users to authenticate to a server before seeing rights-protected search results.

To modify default security settings, do the following:

1 From the Web Search Manager home page, click Security under Default Settings.

2 In the Basis for Authorization drop-down list, choose from the following options:

- Allow All means that although the Login button appears on the default search page, no authentication will be required to view information. All results, whether contained in public or private directories, are returned. Web Search will not ask who the user is. This doesn't mean that if information is contained in an eDirectory protected folder that the user will be able to click the link in the results page and be given access.
- Allow Public means that private content will not be returned in the search results.
- User Login means that depending on what you select under Unauthorized Hits Filtered By, unauthorized search results will be filtered out either by a results template or by the NetWare Web Search engine.

IMPORTANT: These settings apply only to file system indexes and to the server where you have Web Search Server installed.

3 Under Connection Settings, click Yes next to Require HTTPS if you want to protect usernames and passwords as they are sent across the network or Internet.

4 Enter a number (in minutes) in the Auto-logout Time field to direct Web Search to log users out who have been idle for the specified period of time.

5 If you need to change the authentication realm string, enter it in the Authentication Realm String field.

HINT: Specifying the Enterprise Server's authentication realm string in this field makes it so that once users authenticate to the Enterprise Server, they won't have to authenticate again when using Web Search to search and access protected information.

Configuring Services Settings

Services Settings are meant for the administrator of the Web Search server and are intended to give her global control over all virtual search servers, including the ability to completely disable searching by turning it off.

They also allow the administrator to control the overall performance of the Web Search Server.

Configuring General Service Settings

General Services Settings affect error log and server settings for all virtual search servers.

To modify general services settings, do the following:

1. From the Web Search Manager home page, click General under Services Settings.
2. Select where you want log results displayed by choosing one of the following options from the Log Errors To drop-down list:

    - File: When this option is selected, you can click View next to the Log Errors To drop-down list and the log results are displayed in your browser.
    - Console: You can also view log results at the NetWare system console by selecting Console, pressing Ctrl+Esc on your server's keyboard, and then pressing the number corresponding to the Tomcat servlet engine.
    - Both: Displays results in both your browser and at the system console.

    HINT: You can access the log file directly by going to *volume*:\searchroot\errors.log.

3. To start a new log file each time you restart the Web Search server, click Yes next to New Log When Services Load.

    HINT: You can also delete the log file at the path specified above. The log file will be recreated on the first instance of a new error, statistics, etc.

4. To limit the size of the log file, enter a file size (in bytes) in the Maximum Log Size field.
5. To limit the number of indexing jobs that can run at the same time, specify a number in the Maximum Number of Active Index Jobs field.
6. In the Default Location of Virtual Search Servers field, specify the path to where you want all virtual search server files to be stored, including index and configuration files.

    HINT: Changing this setting won't move existing sites to the new default location. But all new virtual search servers will be placed here.

7. To direct Web Search to reload configuration files modified manually, outside of Web Search Manager, click Yes next to the Detect Manual Search Server Changes field.

    If you make changes outside of Web Search Manager, such as modifying a configuration or properties file, Web Search will re-read those files as often as you indicate in the Seconds Between Checking for Changes field.

8. In the Seconds Between Checking for Changes field, specify how often Web Search should check for manual changes (changes made outside of Web Search Manager) to the configuration files.
9. To direct Web Search to reload Web Search templates that have been modified, click Yes next to the Detect Template Changes field.

After making a change to a template from within your HTML editing tool and saving it on your server, Web Search will re-read the template as often as you specify in the Seconds Between Template Updates field. That way you can test your changes almost immediately.

10 In the Seconds Between Checking for Template Changes field, specify how often Web Search should reload search, print, results, and error templates.

11 Click Apply Settings.

Configuring Search Services Settings

Search Services Settings let you turn search capabilities on or off and manage debugging and statistics settings.

To modify search services settings, do the following:

1 From the Web Search Manager home page, click Search under Services Settings.

2 To enable search services for all virtual search servers on your Web Search server, click Yes next to Enable Search Service.

3 Under Debug Settings, click Yes next to Enable Search Debugging if you want to keep a log of all searches and query results going to all virtual search servers.

IMPORTANT: We recommend that you use this features only while setting up or troubleshooting your search services because depending on search activity, the log file can grow in size very quickly. While you can limit it's size (see Step 6) it can become ineffective when the limit is reached and no more activities are added to the log file.

4 Select where you want log results displayed by choosing one of the following options from the Log Debug Messages To drop-down list:

- File: When this option is selected, you can click View next to the Log Debug Messages To drop-down list and the log results are displayed in your browser.
- Console: You can also view log results at the NetWare system console by selecting Console, pressing Ctrl+Esc on your server's keyboard, and then pressing the number corresponding to the Tomcat servlet engine.
- Both: Displays results in both your browser and at the system console.

5 To start a new log file each time you restart the Web Search server, click Yes next to New Log When Services Load.

6 To limit the size of the log file, enter a file size (in bytes) in the Maximum Log Size field.

7 Under Statistics Settings, click Yes next to Enable Search Statistics Logging if you want an updated log file containing statistics about searches performed against all virtual search servers on your Web Search server.

8 In the Seconds Between Statistics Updates field, enter a number (in seconds) that should elapse between updates of the statistics log file.

9 For the next three fields, follow Step 4, Step 5, and Step 6 above.

10 In the Log Error If Search Time Exceeds field, enter a number (in seconds) before Web Search should record the current search as exceeding the specified time limit on the statistics display. This appears as the Limit portion of the statistics display.

11 Click Apply Settings.

Configuring Print Services Settings

To modify print services settings, do the following:

1. From the Web Search Manager home page, click Print under Services Settings.
2. To enable print services for all virtual search servers on your Web Search server, click Yes next to Enable Print Service.
3. Under Debug Settings, click Yes next to Enable Print Debugging if you want print debugging turned on.

   IMPORTANT: We recommend that you use this features only while setting up or troubleshooting your search services because depending on search activity, the log file can grow in size very quickly. While you can limit it's size (see Step 6) it can become ineffective when the limit is reached and no more activities are added to the log file.

4. Select where you want log results displayed by choosing one of the following options from the Log Debug Messages To drop-down list:
   - File: When this option is selected, you can click View next to the Log Debug Messages To drop-down list and the log results are displayed in your browser.
   - Console: You can also view log results at the NetWare system console by selecting Console, pressing Ctrl+Esc on your server's keyboard, and then pressing the number corresponding to the Tomcat servlet engine.
   - Both: Displays results in both your browser and at the system console.
5. To start a new log file each time you restart the Web Search server, click Yes next to New Log When Services Load.
6. To limit the size of the log file, enter a file size (in bytes) in the Maximum Log Size field.
7. Under Statistics Settings, click Yes next to Enable Print Statistics Logging if you want to an updated log file containing statistics about print requests performed on your Web Search server.
8. In the Seconds between Statistics Updates field, enter a number (in seconds) that should elapse between updates of the statistics log file.
9. For the next three fields, follow Step 4, Step 5, and Step 6 above.
10. In the Log Error If Print Time Exceeds field, enter a number (in seconds) before Web Search should record the current print job as exceeding the specified time limit on the statistics display. This appears as the Limit portion of the statistics display.
11. Click Apply Settings.

What's Next

**

4 Designing Your Search Solution

NetWare® Web Search Server is a search service solution designed for adding powerful search capabilities to individual Web sites. It is not intended to index the entire Internet. But because many Web sites are comprised of multiple Web servers located across an enterprise, Web Search was designed to be able to index hundreds, even thousands, of Web sites as part of a single search solution.

In addition to enterprise search solutions, Web Search can also be set up to host multiple, independent virtual search servers, all from a single NetWare 6 server. (See "Becoming a Search Service Host" on page 27.)

This chapter provides suggestions for creating a search solution that fits your particular needs and circumstances. If you are interested in hosting professional search services, you might want to read the section "Becoming a Search Service Host" on page 27.

Components of a Virtual Search Server

A virtual search server is a fully functioning, self-contained search service created for a particular audience, such as a department, organization, or a specific group of customers.

A virtual search server typically contains its own indexes, log files, administration interface, search and print templates, scheduled events, virtual search server name, and an optional alias.

Providing search services involves creating one or more *virtual search servers*.

NOTE: Users cannot search more than one virtual search server at a time. However, a virtual search server can contain indexes created from content on multiple Web sites or file servers.

Taking the time to plan your search service strategy can save you time and money and improve the quality of your service.

When you create a new virtual search server, you create an independent search service, meaning that it is self-contained and doesn't depend on, or interact with, other virtual search servers.

Each virtual search server that you create typically contains one or more of the following components:

- *Indexes*: Files that hold key words and associated URLs of Web sites or file server content that have been indexed, or *crawled*.
- *Themes*: When applied, a theme instantly adds a common look and feel to your search page, search and print results pages, and response and error message pages.
- *Search and Print Results Templates*: Templates that become populated with the results of a search and then are displayed to the user. Depending on which templates are used, the level of detail displayed in search and print results varies.
- *Scheduled Events*: Index management, such as updating or regenerating, can be automated to occur at specific intervals using the Scheduling feature.

Deciding If You Need More Than One Virtual Search Server

To determine if you need more than one virtual search server, answer the following questions:

- ❑ Do you want to host search services for multiple, independent organizations?
- ❑ Do you want to consolidate multiple NetWare 5.1 Web Search servers onto a single machine?
- ❑ Do you need to prevent users from being able to search across multiple indexes at the same time?

If you answered *yes* to any of these, you will likely need to create more than one virtual search server. For information, see Chapter 5, "Creating and Managing Virtual Search Servers," on page 29.

Synchronizing Data Across Multiple Web Search Servers

If you are running multiple installations of Web Search across several NetWare servers, you can simplify management tasks, such as re-generating indexes and sharing configuration settings, by configuring all of your servers into a single Web Search Synchronization cluster.

Web Search Synchronization lets you designate one Web Search server as the *search master* from which updated indexes, templates, and configuration settings are systematically sent out to all other Web Search servers defined as part of a Web Search Synchronization cluster. Each of the other servers must be configured to receive updates from the search master.

Web Search Synchronization offers several key benefits:

- Saves system resources on all other machines because they do not have to regenerate indexes themselves.
- Lets an administrator manage all other Web Search servers from a single interface.
- Regenerates indexes nightly on the search master and then pushes them out to all other Web Search servers.
- Offers failover benefits by hosting Web Search indexes, templates, and configuration settings across multiple servers; if one server goes down, your users will not know the difference.

IMPORTANT: Search Server Synchronization is different from Cluster Services. Novell Cluster Services groups two or more NetWare servers together for the sake of redundancy and for ensuring server availability. A Web Search Synchronization cluster synchronizes indexes, templates, and configuration settings between two or more Web Search servers, thus reducing the administrative work of updating the same index across several Web Search servers. (See "Using Web Search in a NetWare Cluster" on page 26.)

To enable Web Search Synchronization, you need to do the following:

1. Designate one of the Web Search servers as the search master and configure it to send updates to other Web Search servers by identifying each of the other servers to be included in the cluster.
2. Configure which virtual search servers and indexes on the search master will be included in the cluster.
3. Configure each Web Search server in the cluster to receive updates from the search master.
4. Configure each virtual search server that you want included in the cluster to receive data. Depending on your purpose, you might have to create matching virtual search servers.

Setting Up the Search Master

To set up the search master:

1. From the Global Settings page of Web Search Manager, click Cluster under Services Settings.
2. Under General Cluster Settings, click Yes next to Will This Machine Send Cluster Data.
3. Click No next to Will This Machine Receive Cluster Data.

   NOTE: If you want the search master to receive updates from another server in the cluster, you can do so by checking Yes next to Will This Machine Receive Cluster Data.

4. Because this is the search master and will not be receiving cluster data, click No next to Require Admin Authorization When Receiving Cluster Data.
5. **(Optional)** To ensure a secure connection with the other Web Search Servers, click Yes next to Require HTTPS for All Cluster Communications.
6. In the Maximum Number of Transmission Attempts, enter the number of times the search master should attempt to transmit data to an unresponsive Web Search Server in the cluster before quitting.
7. In the Number of Seconds Between Transmission Attempts, type the number of seconds Web Search should wait between each transmission attempt specified in the Maximum Number of Transmission Attempts.
8. **(Optional)** Under Cluster Logging Settings, click Yes next to Enable Cluster Logging if you want Web Search to log cluster-related errors.
9. **(Optional)** Click the Log Cluster Messages To drop-down box to select where log data should be recorded.

   Select File if you want the data stored in a file on your server, or click Console if you want the data displayed on the NetWare system console. Select Both to have the date written to both a file and to the system console. You can also click View to see the latest log data in your Web browser.

10. **(Optional)** Click Yes next to New Log When Cluster Manager loads to start a new logging session when the cluster manager servlet is loaded (or restarted).

    The contents of the current log are replaced with the new data.

11. (Optional) In the Maximum Log Size (Bytes) field, enter the maximum file size of the log file before a new one is started.

    The number you specify here is divided evenly between two log files. For example, if you specify 30000, each log file would allow up to 15000 bytes of logged data. This would ensure that you would always have at least 15000 bytes of logged cluster data.

Defining a Web Search Synchronization Cluster

You must identify each of the Web Search Servers and their virtual search servers to be included in the cluster. To do so, you must know the URL of each server to be included, as well as the names of each virtual search server on each of the Web Search servers that you want included.

Also, if you have configured any of the participating servers to require administrator authorization when being accessed by the search master, you must provide their administrator usernames and passwords (add xref) .

1. In the Cluster *number* Name field, enter a name for the new cluster.

2. In the Server number URL field, enter the URL to a Web Search Server that you want included in the cluster, and specify the name of the virtual search server to be included.
3. **(Optional)** If you enabled (or will enable) the Require Admin Authorization feature on this Web Search Server, enter the admin username and password in the appropriate fields.
4. **(Optional)** To add another server to the cluster, click Add Server and repeat Step 1 through Step 3.
5. Click Apply.
6. To add additional clusters, click Add New Cluster.

Setting Up Search Clients to Receive Updates

(Under development.)

Once the search master is configured, you need to set up each of the other Web Search servers in the search cluster to receive updates from the search master. You must make the same changes on every server to be included in the cluster.

E-mail Feature

(Under development.)

You can specify Web Search to notify you by e-mail when an error occurs with one of your indexes. For example, if an older index is being used when there is a newer index available, Web Search will send an mail indicating the problem. Or if a hard drive runs out of space, Web Search sends a message indicating the problem.

(way of sending an email to whomever indicating an error or warning; an admin feature to keep on top of issues that can occur; file size, old date on an index, or in other words, an older index than the one that is on the server is being used; index failure (harddrive out of space, machine down, whatever might make it fail); cluster push scenario, where a new index is being pushed out to other servers in cluster and then activated; more might be implemented here)

Using Web Search in a NetWare Cluster

Installing and configuring NetWare Web Search Server with Novell Cluster Services™ requires three primary steps:

1. Setting up your clustered environment, including required Web services components.
2. Installing Web Search Server to the same shared volume from each of your clustered servers.
3. Creating virtual search servers using the DNS name and IP address of your cluster server.

Step One: Setting Up Your Clustered Environment

Before NetWare Web Search Server will run in a clustered environment, you must install and configure the following components on a server in your cluster.

❑ Novell Cluster Services

See "Installation and Setup".

❑ Apache Web Server

See "Apache with Novell Cluster Services".

- ❑ Tomcat Application Server

  See "Tomcat and Novell Cluster Services".

Installing Web Search Server to a Shared Volume

Install Web Search Server (using the NetWare 6 Installation CD) to the same shared volume in your clustered environment. For information about setting up a shared volume in a clustered environment, see "Cluster Enable Pools and Volumes" in *Novell Cluster Services Overview and Installation*.

Installing Web Search Server to each server in your cluster allows the installation program to correctly register Web Search with each of your server's Web and application servers.

Installing it to the same shared volume centralizes all of the required templates, indexes, and configuration settings for use by all servers in your cluster. Each time you create a new virtual search server, default settings and templates are drawn from the shared volume. Future upgrades or patches to the Web Search Server need to be installed only to the shared volume and can be done from any server in your cluster.

Step Three: Creating Virtual Search Servers

You can now begin creating virtual search servers from any of the Web Search Servers in your cluster.

When prompted to provide a name and alias for your new virtual search server, enter the DNS name and IP address (alias) of your cluster server. The Path field can be left blank because it is already known by Web Search Server.

If You Are Not Using a Shared Volume

We recommend that you use a shared volume. However, if for some reason your clustered environment is not utilizing a shared volume, you can install Web Search Server to each of your servers and then use mirroring software to synchronize the data between your servers.

If you do this, each time you upgrade or apply a patch to the Web Search Server, you will need to repeat the upgrade installation or patch on each of the servers in your cluster.

For more information about clustering, see *Novell Cluster Services Overview and Installation*.

Becoming a Search Service Host

Companies who want customers to find information about their products out source this functionality to search service companies to make their information searchable. Figure 2, "Search Service Architecture," on page 9 shows how a search query on the Web site www.digitalairlines.com is sent to a Web Search server on the domain search.digitalairlines.com.

With NetWare Web Search Server, you can offer professional search services for other companies. Using a single installation of Web Search, you can host several virtual search servers simultaneously, which means that you could use the same NetWare server to host search services for several client or customer Web sites.

Getting Started

Once you have thought out your search service strategy, you can begin creating and defining your virtual search servers by referring to Chapter 5, “Creating and Managing Virtual Search Servers,” on page 29.

To learn more about customizing your search service, start by reading Chapter 6, “Understanding Templates,” on page 43. If you are already familiar with Web Search and its search and print templates, you might want to skip to Chapter 8, “Working with Template Variables and Search Parameters,” on page 59.

5 Creating and Managing Virtual Search Servers

This chapter provides detailed information about how to create and manage virtual search servers using NetWare® Web Search Manager.

About Virtual Search Servers

By definition, a virtual search server is a collection of one or more indexes and their related configuration files. Indexes are at the heart of a virtual search server. An index is an optimized binary file that contains keywords found in documents hosted on a Web or file server. Indexes are used by Web Search to return search results to users' Web browsers.

Before creating virtual search servers, particularly large or mission-critical ones, you should carefully plan how to configure your search services. A virtual search server that will be used by a small to medium department in a company requires different planning than a search server that will serve thousands of customers on an Internet site.

For information about how to plan an effective search service, see Chapter 4, "Designing Your Search Solution," on page 23.

Creating a Virtual Search Server

Once you have carefully planned your search service, you can start creating and configuring virtual search servers and begin adding indexes to them.

1. From the Web Search Manager Global Settings page, click Add New Virtual Search Server.
2. In the Name field, enter a new virtual search server name, which is typically the DNS or domain name of your server.

   For more information about virtual search server names, see "Naming a Virtual Search Server" on page 30.
3. In the Alias field, enter a virtual search server alias, which is typically the IP address of your server.

   See "Using the Virtual Search Server Alias" on page 30 for more information about aliases.
4. In the Location field, enter the path to where you want the index and configuration files to be stored.

   HINT: If this field is left blank, Web Search will store the virtual search server files in the /*searchroot*/sites/*sitename* directory. Also, you can store the files on any volume on the server where Web Search is installed, but not on other servers.
5. Click Create.

Naming a Virtual Search Server

When a user sends a search query to the Web Search Server, Web Search must determine which of all of your virtual search servers it should use to handle the incoming search request.

Web Search uses two methods for determining this:

1. Matching the domain name of the search query with the virtual search server names available in Web Search
2. Using the SITE=*searchsitename* query parameter to find matching virtual search server names

For example, in the following search request, Web Search uses the domain name search.domainname1.com as the name of the virtual search server:

```
http://search.domainname1.com/NSearch/SearchServlet?query=find+something
```

This approach requires that your server be set up to recognize the domain name search.domainname1.com. Most servers can be set up to recognize and service multiple domain names in both software and hardware virtual server configurations (see “Setting Up Multiple Web Servers” on page 62).

You could also use an IP address to designate the virtual search server. For multiple virtual search servers, this approach would work only in a hardware virtual server configuration where each virtual search server has its own unique IP address.

Hosting Multiple Virtual Search Servers Using One DNS Name

If you are hosting a search service for two or more customers, you can name each virtual search server according to the organization or company name of each customer and then use the &server query parameter when handling search queries. One of the advantages of using the &site query parameter is that it allows you to use a single DNS name.

For example, suppose your server's URL was searchit.novell.com. If you were setting up search services for a company called Digital Airlines and another company called DemoCity, you could host both services on your single server and then simply include the &server=digitalairlines and &server=democity query parameters within the search forms found on www.digitalairlines.com and www.democity.com.

Queries would be sent from the search forms on each Web site to the URLs corresponding to each virtual search server, as in the following:

http://searchit.novell.com/NSearch/SearchServlet?server=digitalairlines

and

http://searchit.novell.com/NSearch/SearchServlet?server=democity

Using the Virtual Search Server Alias

When defining a virtual search server, you are required to give it a name. But Web Search administrators can also define an alias that can be used when identifying a specific virtual search server during a search request.

An alias name typically follows one of two conventions:

1. *An IP address*: This could be used either in the domain name portion of a URL or be included in a search query using the &server query parameter. Using an IP address in place of a domain

name to select a virtual search server only works in a hardware virtual server configuration where each search server has its own unique IP address.

2. Any other numeric or textual value that can be passed as the value of the &server query parameter.

For most virtual search servers, the best choice for a search server name and alias is the Web server's domain name and IP address.For more information about creating software and hardware virtual servers on the NetWare Enterprise Web Server, see "Setting Up Multiple Web Servers" on page 62.

Storing Virtual Search Server Files

Search server files include a set of index and configuration files for each virtual search server. When you create a new virtual search server, you can specify where you want virtual search server files to be stored, or you can accept the default path which is determined by where you installed the NetWare Web Search Server.

Virtual search server files can be stored on any volume visible to the NetWare server that Web Search is installed on, regardless of which volume your Web Search Server is installed on. This includes SAN storage device volumes.

Disabling or Deleting a Virtual Search Server

(Under development.)

Configuring a Virtual Search Server's Settings

(Under development.)

Once created, you can configure your new virtual search server.

**document Settings--start with Global Settings doc in 01install_config.fm

Configuring General Settings

Changes you make to the query, response, and error log settings affect all newly created virtual search servers.

To modify general default virtual search server settings, do the following:

1 From the Web Search Manager home page, click General under Default Settings.

2 From the Default Query Encoding drop-down list, select an encoding that represents the character set encoding that most of your user queries will use.

3 In the Maximum Query Duration field, enter the maximum number of seconds before Web Search should end a query, regardless of whether a search has been completed.

   This option is one of several methods for letting you protect your server's resources from processing potential rogue searches, which are sometimes intended to harm your service by consuming server resources.

4 Under Response Settings, select an output encoding from the Default Encoding for Response Pages drop-down list.

This setting specifies the encoding Web Search should use when responding to user queries using the search and print results templates, and the error and response messages templates.

5. Enter the maximum number of queries in the Refuse Queries if Potential Hits Exceed field to cancel the processing of search results that might take a long time to complete.

6. In the Maximum Log Size field, enter the maximum size (in bytes) that Web Search will allow the log file to grow to.

   Depending on the number of visitors that your virtual search server hosts, log files can become large. This setting will protect your system's hard drive resources.

Editing the Stop-Words List

(Under development.)

Configuring Search Settings

To modify default search settings for search features, do the following:

1. From the Web Search Manager home page, click Search under Default Settings.

2. Under Query Results Settings, enter the number of search results in the Default Number of Results to Display field that you want displayed on each search results page.

   For example, if you set this to 25 (which is the default setting) and the number of hits in a return was 200, Web Search would only return 25 hits per search results page at a time.

3. Set a limit on the number of results allowed at one time on the results page by entering a number in the Maximum Number of Results to Display field.

4. Enter the highest number of search results that can be returned to a user query in the Highest Allowed Result Number field.

5. Under Template Settings, enter a path to where your Web Search templates are stored in the Templates Directory field.

   HINT: The default path is *volume*:\searchroot\Templates, but if you have created custom templates or for some reason want to keep your templates elsewhere, specify the path here so that Web Search knows where the templates are.

6. From the Default Encoding for Templates drop-down list, select the character set that your templates are written in.

   This value will be used even with templates that do not specify an encoding. Encodings found in templates that do not match the encoding you specify here will override this encoding.

7. In the Default Search Page Template field, enter the filename of the search page template you want to use.

   If you have created a custom template and want Web Search to use it as your search page, enter its name in this field.

8. In the Default Search Results Template field, enter the filename of the search results template you want to use.

   If you have created a custom search results template and want Web Search to use it as your default search results page, enter its name in this field.

9. In the Template to Use If No Results Returned field, enter the filename of the template that Web Search should return if no results are found.

10 In the Template to Use If Error Occurs field, enter the filename of the template that Web Search should return if there are errors while processing a user's query.

11 Click Apply Settings.

Selecting Highlighter Colors

(Under development.)

Configuring Print Settings

To modify default print settings, do the following:

1 From the Web Search Manager home page, click Print under Default Settings.

2 Under Print Results Settings, enter the number of print results in the Default Number of Results to Print field that you want displayed on each print results page.

For example, if you set this to 25 (which is the default setting) and the number of hits in a return was 200, Web Search would only return 25 hits per print results page at a time.

3 Set a limit on the number of results allowed at one time on the results page by entering a number in the Maximum Number of Results to Print field.

4 Enter the highest number of search results that can be returned to a user query in the Highest Allowed Result Number field.

5 To limit the size of a print job, specify the largest print job size that Web Search will allow in the Maximum Print Job Size field.

Any users requesting a print job larger than this value will receive a message informing them that their request was too large.

HINT: This is a useful feature to administrators who want to keep down the size of print jobs in their own companies, departments, or organizations.

6 To be notified when a print job exceeds a certain size, enter the print job size in the Print Job Size Warning field.

By default, this message is sent using the ResponseMessageTemplate.html file and is intended as a warning to users that they are exceeding the allowed print job size. It then prompts the user to confirm the print job before continuing.

7 Under Template Settings, enter a path in the Templates Directory field to where your Web Search templates are stored.

HINT: The default path is *volume*:\searchroot\Templates, but if you have created custom templates or for some reason want to keep your templates elsewhere, specify the path here so that Web Search knows where the templates are.

8 From the Default Encoding for Templates drop-down list, select the character set that your templates are written in.

This value will be used even with templates that do not specify an encoding. Encodings found in templates that do not match the encoding you specify here will override this encoding.

9 In the Default Print Results Template field, enter the filename of the print results template you want to use.

If you have created a custom print results template and want Web Search to use it when returning print results, enter its name in this field.

10 In the Template to Use If No Results Returned field, enter the filename of the template that Web Search should return if no print results match a user's query.

11 In the Template to Use If More Information Is Needed field, enter the filename of the template to be sent back to users whose print jobs exceed the size you specify in the Print Job Size field. (See Step 6.)

12 In the Template to Use If Error Occurs field, enter the filename of the template that Web Search should return if there are errors while processing a user's print query.

13 Click Apply Settings.

Configuring Index Settings

These settings are intended to make the process of creating indexes even easier by letting you configure common settings as default settings. This saves you time by not making you make the same selections each time you create a new index.

To modify default index settings, do the following:

1 From the Web Search Manager home page, click Index under Default Settings.

2 Select the type of index that you want as the default index type on the Indexing Management page.

3 Check the URLs Are Case Sensitive check box if you want Web Search to recognize URLs that are different only in character case, but are otherwise identical. For example, www.digitalairlines.com verses www.DigitalAirlines.com.

IMPORTANT: By setting this option to No can help Web Search to avoid indexing duplicate information, which can come from indexing URLs that are presented using different cases but actually point to the same information. However, if a Web server being indexed is configured to differentiate between cases, Web Search might leave out content that you want indexed.

4 Check the Crawl Dynamic URLs (URLs Containing '?') check box if you want dynamic content indexed, in addition to static content.

See "About Indexing Dynamic Web Content" on page 40.

5 Enter a number (in bytes) in the Maximum File Size to Index field to keep Web Search from indexing files larger than the number you specify.

6 In the Maximum Time to Download a URL field, enter a number (in seconds) before Web Search automatically skips the indexing of the specified URL.

7 From the Encoding (If Not in META Tags) drop-down list, select the encoding to be used when indexing files that do not contain an encoding specification.

For example, HTML files can specify their encoding with a Content-Type META tag.

8 Click Apply Settings.

Configuring Access to Secure Content

Security settings let you manage access to indexed content by requiring users to authenticate to a server before seeing rights-protected search results.

To modify default security settings, do the following:

1 From the Web Search Manager home page, click Security under Default Settings.

2 In the Basis for Authorization drop-down list, choose from the following options:

- Allow All means that although the Login button appears on the default search page, no authentication will be required to view information. All results, whether contained in public or private directories, are returned. Web Search will not ask who the user is. This doesn't mean that if information is contained in an eDirectory protected folder that the user will be able to click the link in the results page and be given access.
- Allow Public means that private content will not be returned in the search results.
- User Login means that depending on what you select under Unauthorized Hits Filtered By, unauthorized search results will be filtered out either by a results template or by the NetWare Web Search engine.

IMPORTANT: These settings apply only to file system indexes and to the server where you have Web Search Server installed.

3. Under Connection Settings, click Yes next to Require HTTPS if you want to protect usernames and passwords as they are sent across the network or Internet.
4. Enter a number (in minutes) in the Auto-logout Time field to direct Web Search to log users out who have been idle for the specified period of time.
5. If you need to change the authentication realm string, enter it in the Authentication Realm String field.

   HINT: Specifying the Enterprise Server's authentication realm string in this field makes it so that once users authenticate to the Enterprise Server, they won't have to authenticate again when using Web Search to search and access protected information.

Creating Indexes

Web Search creates two types of indexes:

- Crawled: Created as Web Search follows (or crawls) hypertext links until it reaches a dead end. Web Search can crawl one or more Web sites, specific areas of a Web site, or specific URLs, even down to a specific filename.
- File System: Created as Web Search indexes content on a file server. Web Search can index one or more paths on multiple volumes, including Storage Area Network (SAN) storage devices.

There are two forms you can use to create an each type of index: the standard form and the advanced form.

For example, the Define Crawled Index is the standard form for creating a crawled index. But the Define Crawled Index (Advanced) form offers more options than the standard form, including options that override default virtual search server settings. Both methods are described in the following sections.

Searching across Multiple Indexes

Web Search can search across multiple indexes within a single virtual search server. However, searching a single index is generally faster than searching across multiple indexes.

HINT: While you can search across multiple indexes within a single virtual search server, you cannot search across multiple virtual search servers.

Restricting Search Results to Specific Areas

You can restrict search results to specific areas of your file or Web server in the following ways:

- Using multiple indexes and using the &collection=index_name query parameter.
- Using a single index, restrict results to certain URL paths using the &filefilter=path query parameter.
- Using a single index, restrict results to certain values in document fields by including / fieldname=value with either the query=value or filter=value search parameters.

HINT: Using the last option requires that indexed documents contain summary fields such as META tags. This option works for almost any file format that contains document summary fields, including HTML, XML, PDF, Word*, and WordPerfect*.

For information about preventing Web Search from indexing specific content, see "Excluding Documents from Being Indexed" on page 49.

Defining a New Crawled Index

Using the Define Crawled Index Page

1. From the Web Search Manager Global Settings page, click Manage in the row of the virtual search server that you want to work with.
2. Under Define a New Index, click New Crawled Index > Define Index.
3. In the Index Name field, enter a name for your index.

   HINT: A name can be a word, phrase, or a numeric value. If the virtual search server you are working on contains, or will contain, a large number of indexes, you might want to utilize a numbering scheme to help you manage multiple indexes more effectively. But keep in mind that the name you enter here appears on the default search page. So you might want to choose a name that can be understood by users of your search services.
4. Under Web Sites to Crawl, type the URL of the Web site that you want indexed.

   You can enter just the URL, such as www.mycompany.com, or you can also append a complete path, down to the file level, such as www.mycompany.com/*path*/index.html.
5. If desired, add another URL.
6. To add additional URLs, click Add More URLs.
7. Click Apply Settings.

Using the Define Crawled Index (Advanced) Page

The Define Crawled Index (Advanced) page offers some additional options beyond those available in the standard Define Crawled Index page. Changes made using this page will override default virtual search server settings.

1. From the Web Search Manager Global Settings page, click Manage in the row of the virtual search server that you want to work with.
2. Under Define a New Index, click New Crawled Index > Define Index.
3. On the Define Crawled Index page, click Advanced Index Definition.
4. In the Index Name field, enter a name for your new index.

   HINT: A name can be a word, phrase, or a numeric value. If the virtual search server you are working on contains, or will contain, a large number of indexes, you might want to utilize a numbering scheme to help you manage multiple indexes more effectively. But keep in mind that the name you enter here appears on the default search page. So you might want to choose a name that can be understood by users of your search services.

5. In the Index Description field, enter an optional description of the index to be created.
6. Under Web Sites to Crawl, enter the URL of the Web site to be indexed.

   HINT: If you enter a filename at the end of the URL, then just that file will be indexed.
7. In the Subdirectories to Exclude text box, type the directories that you want Web Search not to index.

   For example, /marketing or /sales/doc.
8. To direct Web Search to include or exclude specific file types, click Extensions to Include or Extensions to Exclude and then enter the extensions, separating each one with a single space, such as HTM PDF TXT.
9. To add additional URLs, click Define More Web Sites.
10. To delete a URL, click Remove Web Site.
11. In the Additional URLs text box, enter any other URLs that you want indexed.

    For example, www.mycompany.com/marketing.

    This allows you to specify additional pockets of information found on other Web sites, but not include all of the content of those sites to your searches.

    HINT: When Web Search encounters links found in the pages of Additional URLs that point to pages specified in Web Sites to Crawl, Web Search follows those links. All other links that go outside of Web Sites to Crawl are not followed.
12. Under Additional Settings, enter the absolute path to where you want the index files stored in the Location of Index Files field.

    For example, *volume*:\searchroot\\sites\\mysites.

    By default, index files are stored at *volume*:\searchroot\\sites\\default\\indexes\\.

    HINT: Changes made to Additional Settings override Default Settings.
13. From the Encoding (If Not in META Tags) drop-down list, select the encoding to be used by files being indexed that do not contain an encoding specification.
14. In the Maximum File Size to Index field, enter the maximum file size (in bytes) that Web Search should index.

    Files exceeding this size will not be indexed and therefore, will not be included in search results.
15. In the Maximum Time to Download a URL field, enter a number (in seconds) before Web Search automatically skips the indexing of the specified URL.
16. To direct Web Search to pay attention to case of filenames and directory names, click Yes next to URLs that are Case Sensitive.
17. To direct Web Search to crawl dynamic content (URLs containing the question mark [?]), click Yes next to Crawl Dynamic URLs.

    HINT: For more information about indexing dynamic content, see "About Indexing Dynamic Web Content" on page 40.

Once you define an index, you must generate it to make it searchable. See "Generating Indexes" on page 39.

Defining a New File System Index

Using the Define File System Index Page

1. From the Web Search Manager Global Settings page, click Manage in the row of the virtual search server that you want to work with.
2. Under Define a New Index, click New File System Index > Define Index.
3. In the Index Name field, enter a name for your index.

   HINT: A name can be a word, phrase, or a numeric value. If the virtual search server you are working on contains, or will contain, a large number of indexes, you might want to utilize a numbering scheme to help you manage multiple indexes more effectively. But keep in mind that the name you enter here appears on the default search page. So you might want to choose a name that can be understood by users of your search services.
4. In the Server Path to be Indexed field, enter the absolute path to the folder containing the information that you want indexed.

   For example, SYS:\SALES\REPORTS.
5. In the Corresponding URL Prefix field, enter the URL that should be used by the search results page to access the individual files.

   For example, /SALES.

   HINT: For information about defining a URL prefix in the NetWare Enterprise Web Server, see “Setting Additional Document Directories” on page 54.
6. To add additional paths, click Add More Paths.
7. Click Apply Settings.

Once you define an index, you must generate it to make it searchable. See “Generating Indexes” on page 39.

Using the Define File System Index (Advanced) Page

1. From the Web Search Manager Global Settings page, click Manage in the row of the virtual search server that you want to work with.
2. Under Define a New Index, click New Crawled Index > Define Index.
3. On the Define File System Index page, click Advanced Index Definition.
4. In the Index Name field, enter a name for your new index.

   HINT: A name can be a word, phrase, or a numeric value. If the virtual search server you are working on contains, or will contain, a large number of indexes, you might want to utilize a numbering scheme to help you manage multiple indexes more effectively. But keep in mind that the name you enter here appears on the default search page. So you might want to choose a name that can be understood by users of your search services.
5. In the Index Description field, enter an optional description of the index to be created.
6. In the Location of Index Files field, enter the absolute path to where you want the index files stored.

   For example, SYS:\NSearch\sites\mysites.

   By default, index files are stored at *volume*:*searchroot*\sites*site_name*\ indexes\.
7. From the Encoding (If Not in META Tags) drop-down list, select the encoding to be used when indexeing files that do not contain an encoding specification.

For example, HTML files can specify their encoding with a Content-Type META tag.

8. In the Maximum File Size to Index field, enter the maximum file size (in bytes) that Web Search should index.

   Files exceeding this size will not be indexed and therefore, will not be included in search results.

9. Under Path Information, type the absolute path to the folder containing the information that you want indexed in the Server Path field. For example, SYS:\SALES\REPORTS.

10. In the Corresponding URL Prefix field, enter the URL that should be used by the search results page to access the individual files.

    For example, /SALES.

    HINT: For information about defining a URL prefix in the NetWare Enterprise Web Server, see "Setting Additional Document Directories" on page 54.

11. To exclude specific subdirectories from being indexed, enter their relative paths in the Subdirectories to Exclude field.

12. To direct Web Search to include or exclude specific file types, click Extensions to Include or Extensions to Exclude and then type the extensions, separating each one with a single space, such as HTM PDF TXT.

13. To add additional paths, click Define More Paths.

14. To delete a path, click Remove Path.

15. Click Apply Settings.

Once you define an index, you must generate it to make it searchable. See "Generating Indexes" on page 39.

Generating Indexes

Once you define an index, you must generate it before it can be used for searching. Generating an index is the actual process where Web Search Server examines file server or Web server content, gathers keywords, titles, and descriptions and then includes them in the index.

1. From the Web Search Manager Global Settings page, click Manage in the row of the virtual search server that you want to work with.

2. Click Generate in the Action column of the index that you want to work with.

   The Active Jobs screen indicates the status of the current indexing jobs. When there is no current index job, the status page will read `No indexing jobs are currently running or defined`.

3. To cancel the current indexing jobs, click Cancel in the Status column.

You can direct Web Search to automatically update your indexes on specific dates and at specific times by scheduling events. For more information, see "Automating Index and Server Maintenance" on page 41.

Managing Existing Index Files

Once created, an index can then be edited or deleted. You can also view an index's log file. (See "Working with the Log File" on page 40)

Editing an Index

1. From the indexing Management page, click Edit in the Action column of the index you want to work with.
2. Make any of the changes you need to and then click Apply Settings.

   HINT: If you used the Advanced page to create the index, it will appear automatically. However, you can also click Advanced Index Definition to make advanced changes to an index you created using the standard Index Definition page.
3. If you added new paths or URLs or modified any of the existing ones, you should regenerate the index to include the new content.

Deleting an Index

1. From the indexing Management page, click Delete in the Action column of the index you want to delete.
2. In the Confirm Deletion of *indexname* page, click Delete Index to proceed, or click Cancel Deletion.

   WARNING: Once an index has been deleted, it cannot be restored. You must generate a new index.

Working with the Log File

The purpose of the log file is to help you identify any errors (and their possible causes) in performance during an indexing job.

In addition to reporting when the indexing job started and stopped, it also lists all files that were indexed, files that could not be found but were linked to, and even errors that might have occurred during the indexing process.

To view an index's log file, do the following:

1. Click View Log in the Action column of the index that you want to work with.
2. Review the contents of the log file and then either click your browser's Back button to return to the indexing Management page, or click Management in the left frame of the Web Search Manager.

About Indexing Dynamic Web Content

Much of the content on the World Wide Web is static HTML, which means that after a static Web page is created, it remains the same until someone updates it. By contrast, many newer Web pages are created by Web applications, including servlets, Java Server Pages (JSP), Common Gateway Interfaces (CGI), and Pearl Scripts, and are usually created in response to user input.

An example of dynamic Web content is an eCommerce Web page where items to be purchased are stored in a virtual shopping cart, the total cost is updated as users add or remove items from their shopping cart.

Because the content changes regularly, many search engines don't index dynamic content.

NetWare Web Search includes the ability to index dynamic content. The URL of dynamic Web content typically includes a question mark (?). You can direct Web Search to index these URLs by setting the Crawl Dynamic URLs option to Yes. You could then create a scheduled event that regenerates the specified indexes every few minutes.

Automating Index and Server Maintenance

You can eliminate a lot of manual work in keeping indexes up to date by using Web Search's index scheduling feature. Because the Web and file content you have indexed will eventually change, you can direct Web Search to update your indexes on specific dates and at specific times or intervals.

Adding a Scheduled Event

1. After selecting a search server from the Virtual Search Server List, click Scheduling in the left frame of Web Search Manager.
2. Click Add Event.
3. Specify the month, days, days of the week, or time (in hours and minutes) when you want Web Search to run the event.

   HINT: To select multiple dates and times, hold down the Ctrl key and click all of the items you want added. To select consecutive items, click the first item and then hold down the Shift key and click the last item.
4. Select the type of operation you want performed on your indexes.
   - Update: Web Search identifies new content on Web or file servers and updates the index.
   - Optimize: Web Search improves searching performance by removing unnecessary content and making the index file more compact.
   - Regenerate: Web Search replaces the existing index with a newly generated one.
5. In the Perform Operations On column, determine whether you want the chosen operation performed on all indexes or only on specified ones.

   HINT: If you have large indexes, you might consider creating multiple events that update your indexes at varied times. Doing so will minimize CPU utilization. By default, Web Search supports up to 5 simultaneous indexing sessions. All other indexes will wait until a previous index job has completed. You can control the number of simultaneous indexing jobs from Services Settings. (See "Configuring Services Settings" on page 20.)
6. Click Apply Settings.

Editng or Deleting an Event

1. After selecting a virtual search server from the Virtual Search Server List, click Scheduling in the left frame of Web Search Manager.

   HINT: If no events have been scheduled, refer to the procedure above for adding a scheduled event.
2. To edit a scheduled event, click Edit in the row of the event you want to modify.
3. Make the desired changes and click Apply Settings.
4. To delete a scheduled event, click Delete in the row of the event you want to delete.
5. Click Delete Event to confirm the deletion, or click Cancel Deletion.

Access Control to Search Results

(Under development.)

In NetWare 6, Web Search depended on Enterprise Web Server NLMs to authorize each search result for a particular user for those documents managed by eDirectory. Web Search Server now depends on the rights engine built into NetWare 6.5. In addition to other enhancements, this allows user privileges to entire indexes rather than to each search result, which improves the over all speed at which search results requiring authentication are returned to the user.

(**Insert procedure.)

Backing Up Your Virtual Search Server Files

As with any valuable data, you should make sure that your virtual search server files are backed up. At minimum, you should back up your index files, which by default are stored at *volume*:\searchroot\sites.

However, if you have customized templates, you might also want to back them up. By default, they are stored at *volume*:\searchroot\templates.

6 Understanding Templates

NetWare® Web Search Server utilizes templates to generate search forms and search and print results as well as user feedback such as error or response messages.

A template is an HTML document containing one or more Web Search Server variables. Template variables are used to produce dynamic results when a user performs a search on the virtual search server you have defined.

Templates can be shared across virtual search servers or each virtual search server can point to its own set of templates.

This chapter describes how templates work and discusses the default NetWare Web Search Server templates that are included. To learn how to customize the default templates, see Chapter 9, "Customizing Your Search Solutions," on page 77, and Chapter 8, "Working with Template Variables and Search Parameters," on page 59.

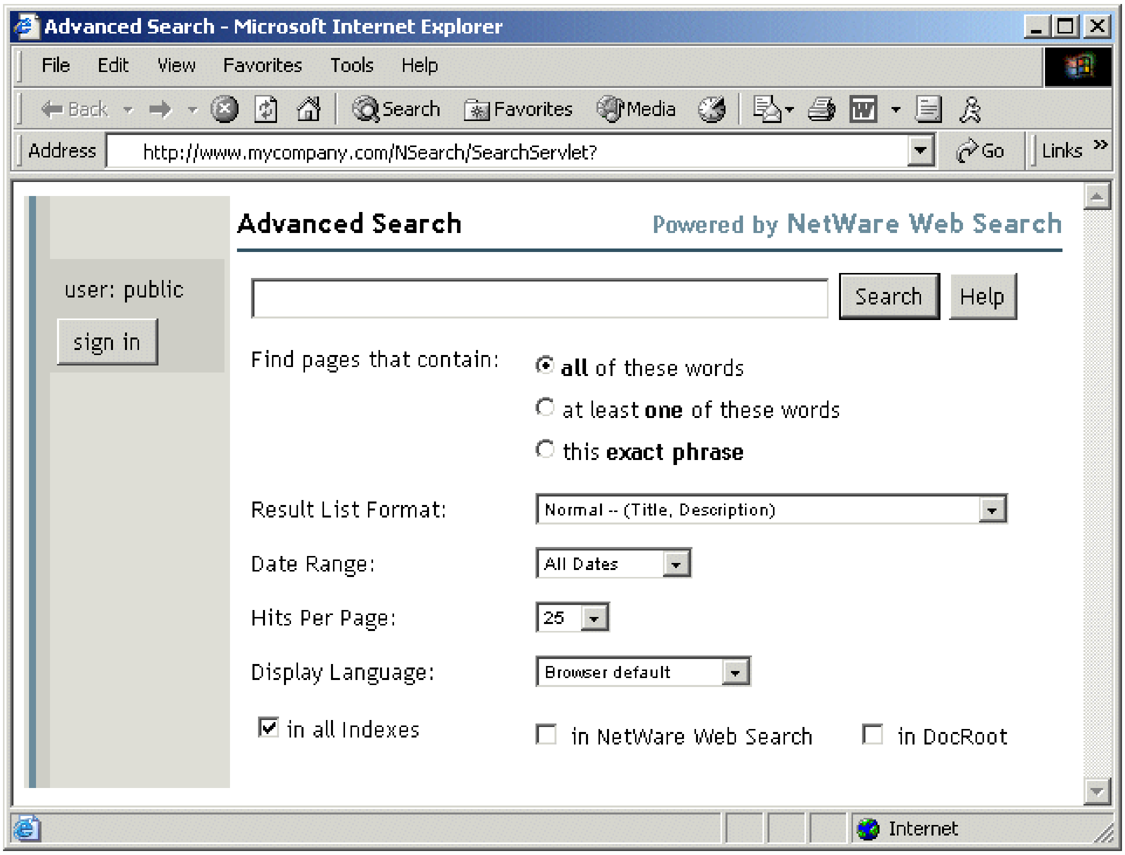
How Templates Work

As defined above, a template is normally an HTML document containing one or more Web Search Server variables. When users search your virtual search server, they use a Web browser to access the search form template. See Figure 4, "The NetWare Web Search Form As It Appears in a Web Browser," on page 44.

The Search form template, SearchTemplate.html, is stored (by default) on *volume*:\searchroot\TEMPLATES. This path might be different if you chose to install Web Search in another directory.

For more information about customizing templates, see "Customizing the Look and Feel of Search Server Content" on page 11.

Figure 4 The NetWare Web Search Form As It Appears in a Web Browser



http://www.mycompany.com/NSearch/SearchServlet?

Advanced Search

Powered by NetWare Web Search

user: public

Find pages that contain:
● **all** of these words
○ at least **one** of these words
○ this **exact phrase**

Result List Format: Normal -- (Title, Description)

Date Range: All Dates

Hits Per Page: 25

Display Language: Browser default

☑ in all Indexes
□ in NetWare Web Search
□ in DocRoot

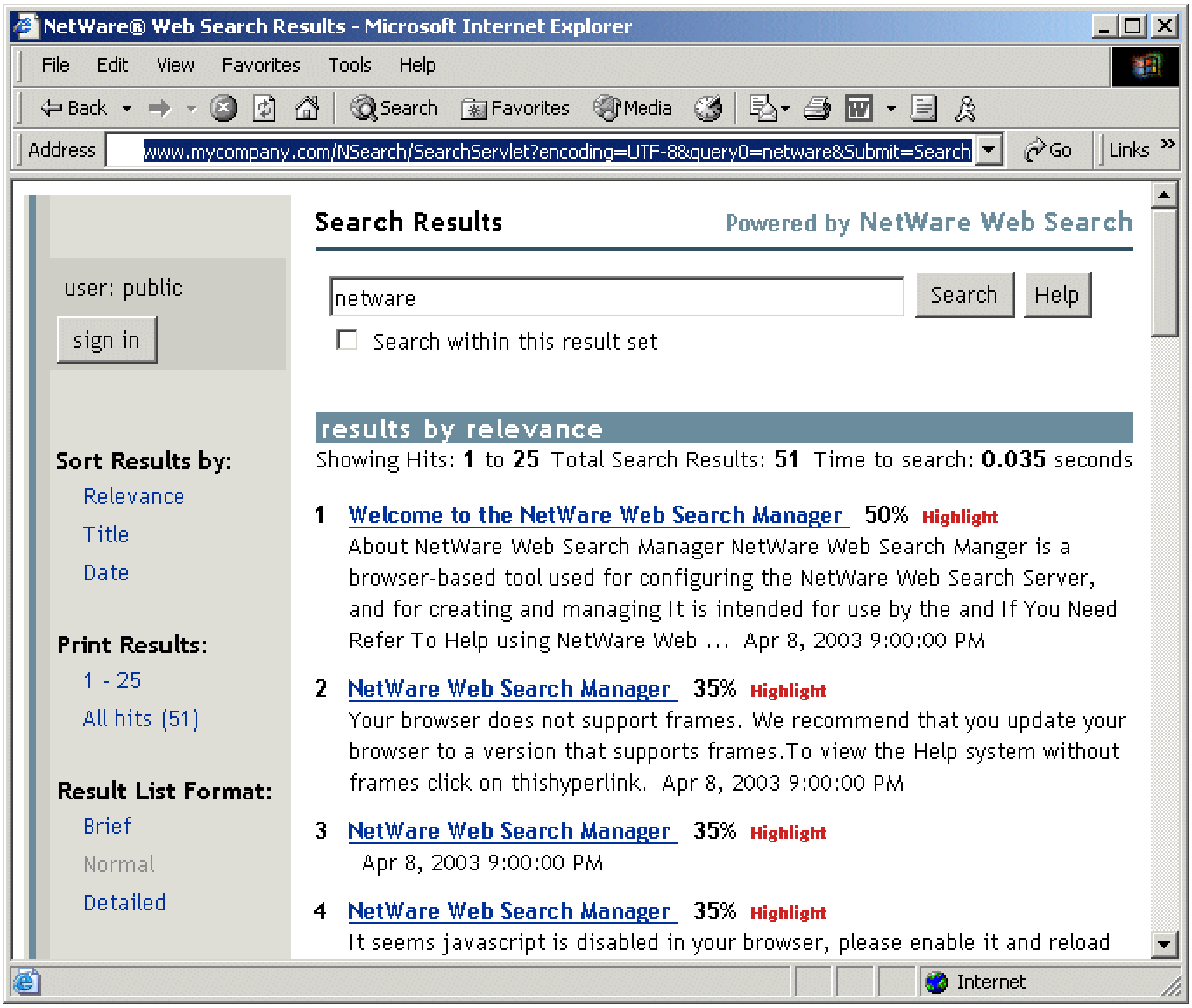
The Web Search form is used to capture user input, select available indexes, and then return the results in either a search or print results template, which appears to the user in a dynamically updated HTML document.

Search result templates display hits according to user selections on the search form. For more information about these search result templates, see Table 2 on page 46.

There are also search and print templates for several different languages. For a discussion about creating templates for international languages, see Chapter 10, “Internationalizing Your Search Solution,” on page 81.

After a query is submitted and results are found, Web Search populates a results template with all relevant information for each search result. (See Figure 5 on page 45.)

Figure 5 A Search Results Page Produced by the Search Results Template, ResultListTemplate.html.



Address: www.mycompany.com/NSearch/SearchServlet?encoding=UTF-8&query0=netware&Submit=Search

netware

□ Search within this result set

results by relevance

Showing Hits: **1** to **25** Total Search Results: **51** Time to search: **0.035** seconds

1 Welcome to the NetWare Web Search Manager 50% Highlight
About NetWare Web Search Manager NetWare Web Search Manger is a browser-based tool used for configuring the NetWare Web Search Server, and for creating and managing It is intended for use by the and If You Need Refer To Help using NetWare Web ... Apr 8, 2003 9:00:00 PM

2 NetWare Web Search Manager 35% Highlight
Your browser does not support frames. We recommend that you update your browser to a version that supports frames.To view the Help system without frames click on thishyperlink. Apr 8, 2003 9:00:00 PM

3 NetWare Web Search Manager 35% Highlight
Apr 8, 2003 9:00:00 PM

4 NetWare Web Search Manager 35% Highlight
It seems javascript is disabled in your browser, please enable it and reload

You can also customize the search form to include additional parameters that allow you to offer more options to your users for more accurate searching.

Exploring the Default Search and Print Templates

NetWare Web Search Server includes several default templates used to create search forms and to format, display, and print search results for users.

You can use the templates as they are or you can modify them to look and feel how you want them to. You can also create as many additional templates as you need or replace the default templates with your own templates.

NetWare Web Search includes the five template categories:

- Search Page Templates
- Search Result Templates
- Print Result Templates
- Error Message Template

- Response Message Template

The templates are stored at *volume*:\searchroot\TEMPLATES.

Search Page Templates

NetWare Web Search includes two search page templates that are used to generate a search page, as described in the following table.

Table 1 Default Search Page Templates

Template Name	Purpose
SearchTemplate.html	Lets users select a variety of options when performing searches and is the default search template used by NetWare Web Search Server.
SearchTemplate.Simple	Similar to SearchTemplate.html, except that this template contains no dynamic indexes.

Search Result Templates

NetWare Web Search includes several ready-made result templates, as described in the following table.

Table 2 Default Search Result Templates

Template Name	Purpose
ResultListTemplate.html	Formats and organizes search results and offers additional sorting functions to the user.
ResultListNoHitsTemplate.html	Indicates when no hits are found during a search and offers users a chance to refine their search.
ResultListTerseTemplate.html	Similar to ResultListTemplate but returns less information, such as dates and titles only.
ResultListVerboseTemplate.html	Similar to ResultListTemplate, but returns more information, such as file date, time, and language. Additional sort options are also provided.

Print Result Templates

From the search results page, users have the option of printing all files matching their search or only those files displayed on the current search results page. When one of these options is selected, the print result templates described in the following is displayed.

Table 3 Default Print Result Templates

Template Name	Purpose
PrintResultTemplate.html	Combines the full contents of each of the files in the print request into a single document which is then displayed in the user's Web browser. A dynamic table of contents is also created. Once the entire content is downloaded to the browser, the browser's print dialog appears.
PrintResultNoHitsTemplate.html	Indicates when no documents are found during a print request.

Error and Response Message Templates

In addition to the print, search, and search result templates, the error and response message templates are returned when an error occurs or when information is needed from the user.

The default response message template is returned to convey a specific message to the user such as "Print job exceeds recommended size limits," typically returned when a user attempts to print more content than the Web administrator has allowed.

The error and response message templates can be found at *volume*:\searchroot\TEMPLATES\ErrorMessageTemplate.html and ResponseMessageTemplate.html.

How Templates Use System Memory

Templates are cached in memory for quick rendering speed. Each template consumes approximately 10 KB.

Similar to the virtual search server cache, templates remain cached in memory until a period of inactivity has elapsed. The template is then dynamically removed from memory until its next use. The first time a template is accessed, therefore, is normally the slowest.

HINT: Too many templates in the template cache can consume a great deal of memory. Try to share templates across sites to minimize the impact on system memory resources.

Working with Additional Languages

NetWare Web Search includes each of the templates described above in each of several languages. Using standard encoding practices, you can internationalize your templates.

Any changes made to the default templates should also be made to the language templates you will use. For a more complete discussion about creating a multilingual search solution, see Chapter 10, "Internationalizing Your Search Solution," on page 81.

7 Optimizing Search Results

There are a number of ways administrators can optimize the performance of their virtual search servers.

Improving Search Results through Intelligent Indexing

You can improve the accuracy of your search results by following these indexing guidelines:

- ❑ When defining and creating your indexes, start with the highest-level Web Site URLs and File System Paths possible.
- ❑ If content is showing up in your search results that you don’t want included, try removing some paths or URLs from your defined indexes. Also, try excluding specific subdirectories that you know or suspect might contain content that you don’t want searched.
- ❑ If you've indexed too many file types and cluttered your search results, try removing file types that you don’t want indexed by using the Extensions to Exclude option on the Define Index page.
- ❑ Use the Robots META tag in your Web site’s content.
- ❑ Exclude documents or specific sections of documents, including headers, footers, and navigation bars.

Excluding Documents from Being Indexed

One way to improve search results is to guard what content is actually indexed, thus clearing a path for relevant information.

Using the Extensions to Exclude Option

You can use the Extensions to Exclude option to direct Web Search to ignore specific file types. For example, if you don’t want Word or PowerPoint documents to be included in search results, you would enter DOC and PPT in the Extensions to Exclude field. When these document types are encountered during and indexing job, Web Search skips over them.

Using the Extensions to Include Option

As mentioned above, you can use the Extensions to Exclude option to direct Web Search to ignore specific file types. However, if you can't specify all of the extensions to exclude, use the Extensions to Include option and specify all acceptable file extensions. A typical list would specify HTM, HTML, PDF, TXT, and DOC.

HINT: When entering extensions in the Extensions to Exclude box, separate each extension by a space or a hard return. Avoid using commas. For example:

```
htm html pdf txt doc
```

Using the Robots META Tag

Another effective way of controlling what Web Search indexes is using the Robots META tag, a tag inserted into the content that is being indexed by Web Search.

When a Web-based search engine encounters a document containing the Robots META tag, the search engine will do as the META tag instructs.

There are several values you can specify in the Robots META tag:

- NOINDEX: Indicates that the document is not to be indexed.
- NOFOLLOW: Indicates that hypertext links in the document are not to be crawled.
- FOLLOWINDEX: Indicates that hypertext links in the document should be crawled.
- ALL: Indicates that the document can be indexed and all links can be crawled.
- NONE: Indicates that the document is not to be indexed and that hypertext links are not to be crawled.

To include the Robots META tag, use this syntax:

```
<META name="Robots" content="value, optional_value">
```

Using the Robots Comment Tag

You can also use the Robots Comment tag to exclude specific sections of HTML documents from your search results. For example, you might not want such sections as repetitive headers, footers, navigation bars, and server-side includes to be indexed.

HINT: You can also place these tags at the top and bottom of all include files so these sections never get indexed when part of a larger document.

To direct Web Search where to begin skipping content while indexing, do the following:

1. At the point in your HTML document where you want Web Search to begin skipping content while indexing, enter the following tag:

   ```
   <!--*Robots NoIndex-->
   ```

2. Just after the content you want skipped, enter the following tag:

   ```
   <!--*Robots Index-->
   ```

3. Save your changes and index (or reindex) the content.

Modifying Document Descriptions Returned in a Search Results List

Web Search returns descriptions of each hit that is listed on the search results page. By default, the following information is returned for each result:

- Description field
- Summary field
- Abstract field
- The first 255 characters of the document (beginning with first heading and skipping links)

The first three fields are taken from the content of META tags in HTML documents or from document summary fields in other document types such as Word* or PDF files. If these tags or fields are not defined, Web Search will try to find the first heading and begin selecting words. If it can't find a heading, then it begins at the top of the document and selects the first 255 relevant display bytes as the description.

Improving the Relevance of Search Results

Web Search utilizes a sophisticated relevance-ranking algorithm. During a search, Web Search considers

- The number of times words appear in a document
- The proximity of words in a multiple word search (the closer the words appear, the more relevant the document will be)
- The order of words in a multiple word search (the exact order of words is more relevant)
- The location of words in a document (specifically words that appear in a META tag, title, body, header, footer, etc.)
- The formatting of words in a document (such as bold, font type and size, etc.)
- Query weighting in a multiple query scenario
- The number of times words occur within an entire index (for example, the word *the* has low relevance)

To illustrate how these criteria work, consider the following examples:

- Words in bold face are more relevant than regular words.
- Words contained in the <Title> tag are more relevant than words contained within the <body> tag.
- Words contained in the Keywords and Description META tags are more relevant than content words.
- Words contained within the <A HREF=> tag used for creating links are less relevant than words outside of this tag.
- A document containing a specified search term multiple times is more relevant than a document that contains the search term only once.
- A word within a 36-point body text is more relevant than within 4-point footer text.
- Documents returned from a query that is weighted at 100% is more relevant than a 50% weighted query. This is normally used in multi-query searches where each query has a specified weight. For example:

  ```
  query0=netware&weight0=100&query1=groupwise&weight1=100
  ```

Using Stop Words Processing

Sometimes users include irrelevant words in their search strings, such as the conjunctions *to and of*. These are referred to as *stop words*. When enabled, the Stop Words feature of Web Search removes all occurrences of stop words from the search string before performing a search.

A set of common stop words are included in Web Search, but you can easily add your own, or even remove the one or more of the ones we have included. (See **xref)

Before a virtual search server can use stop words processing, it must first be enabled on the Web Search server under Global Settings.

To enable Stop Words processing on a Web Search server:

1. From the Global Settings page of Web Search Manager, click General under Default Settings.
2. Under Query Settings, click Yes next to Enable Stop Words Processing.
3. **(Optional)** Click Edit List to modify the default list of stop words. (This is not yet functional in Beta 2--there is no way to edit the list)
4. Click Apply Settings.

To enable stop words processing on a virtual search server:

1. From the Web Search Manager Global Settings page, click Edit in the row of the virtual search server that you would like to enable stop words processing on.
2. Under Settings, click General.
3. Click Yes next to Enable Stop Words Processing.
4. **(Optional)** Click Edit List to modify the default list of stop words. (This is not yet functional in Beta 2--there is no way to edit the list)
5. Click Apply Settings.

Using Best Bets

(Under development.) Best bets is a secondary results list that appears at the top of the search results page and is generated from a special-purpose index, or an index created for the express purpose of generating a best bets results list.

A best bets list can help users find what they are looking for more quickly by bringing the most popular or most important things to the top of the results page.

A special-purpose index typically contains information about the most popular, most recent, or most important documents.

To enable best bets on a Web Search server:

1. From Global Settings, click Search under Default Settings.
2. Click Yes next to Enable Best Bets Search Results.
3. **(Optional)** If you want the Best Bets results to show automatically on the search results page, click No next to Show Best Bets Searches By Default.
4. **(Optional)** In the Maximum Number of Best Bets Results Per Page field, type the maximum number of Best Bets results to be returned on each search results page.
5. Click Apply Settings.

To enable Best Bets on a virtual search server:

1. From Global Settings, click Manage in the row of the virtual search server that you want to modify.
2. Under Default Settings, click Search.
3. Click Yes next to Enable Best Bets Search Results.
4. **(Optional)** If you want the Best Bets results to show automatically on the search results page, click No next to Show Best Bets Searches By Default.

5. **(Optional)** In the Maximum Number of Best Bets Results Per Page field, type the maximum number of Best Bets results to be returned on each search results page.
6. Click Apply Settings.

Using Synonyms To Broaden Search Results

(Under development.)

When enabled, this feature returns documents in the search results that contain synonyms of the user's original search terms. Search results might be less relevant when this feature is enabled, but it will also help identify content that might not otherwise be found due to the choice of search terms. While common synonyms are included, you can specify additional synonyms.

A synonym-derived search result should be slightly less-relevant than the original term. This value indicates how much less relevant a synonym-derived hit is.

Obviously, a document which contains both the original term and one of its synonyms is going to be even more relevant than a document with just the original term

Expands the search results to include synonyms of the original search terms

Synonym-derived search results are slightly less relevant than the original terms

Documents that contain both the original terms and their synonyms will be more relevant

Administrators control the synonym list

Advantages

Users can find documents without knowing the exact terminology

Redirecting A Search

The redirection feature lets you specify key words that are to redirect the user's Web browser to a specific URL. For example, searching for *one net* on www.novell.com redirects your browser to a special page designed to emphasize Novell's One Net strategy, rather than returning results on the search results page.

To enable search term redirection on a Web Search server:

1. From Global Settings, click Search under Default Settings.
2. Under Query Settings, click Yes next to Enable Search Term Redirection.
3. Click Apply Settings.

To create or modify the redirection list:

1. From Global Settings, click Search under Default Settings.
2. Click Edit List to create (or modify) the list of redirection terms and their associated URLs.
3. In the When Search For field, type a search term.
4. In the Go to URL field, type the associated URL.
5. Click Apply Settings. (add additional steps post-beta2)

Helping Users Around the Not Found Message

(Under development.)

If a user's search query results in a *Not Found* message, NetWare Web Search automatically performs a second search, broadening the search to include additional indexes not searched the first time.

Enhancing Search Results

Once results are returned to users, you can further enhance their experience by enabling the Highlighter, Sneak Peek, and Search Within Search features.

Highlighter

(Under development.)

Highlights the search terms and phrases wherever they appear in the actual documents found during the search without affecting the formatting of the original documents. As an administrator, you can specify characteristics of the highlighting, including the markup color. Search terms found in hidden document summary fields and meta tags are also highlighted.

Sneak Peek

(Under development.)

Lets you see a preview of the actual documents listed on the search results page without opening a separate window or leaving the search results page.

Search Within Search

(Under development.)

Lets users perform a second search from within the results of an original search.

Weighted Queries

A weighted query is used anytime you want to modify the order or relevance of certain hits in a user's normal search results list or when you want to add additional search results users might not have identified in their queries.

Web Search allows users to submit more than one query item as part of a single search request.

The following query parameters are combined to identify a single search query item:

&filter#=
&filteroperator#=
&operator#=
&query#=
&weight#=

One use of this feature could be to provide profile-enhanced search requests. For example, the following query returns French product downloads higher up in the search results list but does not eliminate results of any other language downloads:

```
&query0=product+downloads&weight0=100&query1=/language=french&weight1=90
```

This example directs Web Search to perform two completely separate searches. The search results from the two queries are then merged based on the relevance of the individual search results and the weighting of the respective query that produced them.

Another example might be to give the search results from one index more or less relevance than the search results of another index when performing a multiple-index search. For example, the search results from Novell might be more relevant than the search results from Novonyx.

To send multiple query items, these parameters must be grouped using a number (#) at the end of the parameter name so they will be interpreted properly. The numbering should begin at 0 or 1 and increment sequentially for each additional query item.

Ensuring Optimal Search Speed

Once a virtual search server has been accessed, all of its configuration files are read into memory. For speed reasons, the virtual search server remains cached in memory until a period of inactivity has elapsed. The virtual search server is then dynamically removed from memory until its next use. Because of this, the first time a virtual search server is accessed is usually the slowest.

However, there are other factors that can affect the performance of your Web Search services. As with any software, the amount of system resources (CPU, RAM, and hard drive) available affects Web Search Server performance. Web Search speed depends on the following factors:

- System processor speed
- Number of processors
- Amount of system memory (RAM)
- Number of hosted virtual search servers
- Number of indexes within each virtual search server
- Number of files included within each index
- Number of indexes included within each query
- Number of queries performed at one time
- Complexity of users' queries
- Number of search results returned with each results page
- Number of concurrent active indexing jobs
- Other functions being performed by your server

Adjusting any of these values can have a significant impact on the performance of your search services.

As a general guideline, use the fastest CPU possible and include as much RAM as possible. Although the duration of each user query is very short, while it is active it consumes an average of 500 KB of memory. Memory consumption varies widely while the indexer is calculating the final search results list, depending on the number of possible search results.

Also, try to schedule the regeneration of your indexes during off-peak hours. That way, they won't interfere with normal user searches. (See “Automating Index and Server Maintenance” on page 41.)

Making Good Use of Document Fields

A document field is any META tag or document summary field that helps to identify the document’s contents. A document summary field might be a title, heading, or paragraph contained in a TITLE or META tag within an HTML document.

Web Search is designed to take advantage of document fields in order to improve the accuracy, relevance, display information, and speed of search results.

By design, Web Search always indexes all document fields in many document types, including HTML, PDF, Word, WordPerfect, XML, etc. Users can then constrain searches to the contents of any document field.

As a Web Search administrator, you can also use document fields to further restrict search results to certain products, categories, authors, titles, keywords, or any other content belonging to a document field.

To perform a field-restricted search, use the /*fieldname=search_criteria* search operator.

HINT: You might consider sending this information as hidden data using the &filter= query parameter. For example:

```
&filter=/product=netware
```

Searching XML Documents

XML documents provide a tremendous advantage to narrowing search results because of their hierarchical structure and use of multiple document summary fields.

Web Search provides complete hierarchical searching using the fieldname=*search_criteria* operator. For example, you can find information anywhere in the XML document, within any of the title tags, or limit it to within the title tag that is part of the <DOCUMENT><SUMMARY> hierarchy.

The following table shows example uses of the fieldname=*search_criteria* operator when performing a search in XML documents.

Example Values	**Result**
search_criteria	Finds *search_criteria* anywhere in the document.
/<Document*=*search_criteria*	Finds *search_criteria* anywhere within any tag that is part of the DOCUMENT hierarchy.
/<Document<Summary*=*search_criteria*	Finds *search_criteria* within any tag that is part of the <DOCUMENT> or <SUMMARY> hierarchy.
/<Document<Summary<Title=*search_criteria*	Finds *search_criteria* only within the <DOCUMENT><SUMMARY><TITLE> hierarchy.
/<Document*<Title=*search_criteria*	Finds *search_criteria* within any TITLE tag, located at any level within the DOCUMENT hierarchy.

Example Values	Result
/<*<Title=*search_criteria*	Finds the *search_criteria* within any TITLE tag in the document.

Using the &filter Query Parameter

The &filter query parameter allows Web Search administrators to enhance searches by adding hidden, additional query details when users submit a search query. This is an enhancement over previous versions of Web Search, which required that you use Java Script to add additional details to search queries.

The &filter query parameter works just like the &query= parameter and can be used together using the optional number (#) value. For example, if the query parameter was &query0=*search_criteria*, the matching filter parameter would be &filter0=*additional_hidden_search_criteria*. This allows the multiple weighted queries feature to work as designed while allowing administrators to add additional query details to each query.

Unlike the &query parameter, the &filter parameter can be sent multiple times. For example, if users search for software patches, you could include the various products to be searched, which could then improve search time and accuracy:

```
query=software patchesfilter=/Products=Product257filter=/
Products=Product16filter=/Products=Product302
```

The resultant URL might appear as follows, but with the HTTP and domain name prefix:

```
&query=software+patches&filter=%2FProducts%3DProduct257&filter=%2FProducts%3DProduct16&filter
=%2FProducts%3DProduct302
```

NOTE: All &filter operators are combined using default the &operator=*value*, AND. Also, the default Boolean conjunction joining the various filter operators is an OR search. You can change the default Boolean conjunction by using the &filteroperator=# query parameter. The pound sign (#) here acts just like the one used in the #operator=# query parameter.

8 Working with Template Variables and Search Parameters

If you are a developer or are comfortable programming in HTML and working with variables and parameters, you can create an advanced search solution that your users can use to perform complex searches.

Building an advanced search solution involves the use of search and print template variables and search parameters to create or customize search and print templates, and to create or customize one or more search forms.

You must also have used NetWare® Web Search Manager to define and generate one or more indexes.

The Web Search Manager is accessed from NetWare Web Manager. For more information about using NetWare Web Manager, see Chapter 2, "Introducing NetWare Web Manager," on page 17.

Guidelines for Using Variables

Please note the following guidelines when using variables to either customize the default templates, or to create new templates from scratch:

- Case Sensitivity: All variables are case sensitive. Changing case in a variable will cause Web Search to ignore the variable.
- Variable Formatting: All variables must be used exactly as they appear in the tables below. Variables always begin with two dollar signs ($$) next to each other.
- Success of a Variable: The inclusion of a variable does not guarantee that information will be returned after a search is performed. For example, using the $$Author variable might not return the name of a document's author if that information is not included in the META tag of the document.
- Internationalizing Templates: If you want to internationalize your templates, you must create a template for each language you want to support in your search solution. For more information about languages, see Chapter 10, "Internationalizing Your Search Solution," on page 81.

IMPORTANT: In prior versions of NetWare Web Search Server, the term *search site* was defined as a collection of one or more indexes and related configuration files. To avoid confusion with the term Web site, the term was changed wherever it appeared in the documentation and in the variables and parameters. *Search site* is now referred to as *virtual search server*.

New variables and parameters that parallel the term *virtual search server* have been added. Note that they function identically to the prior variable and parameters, and that the old variables and parameters can still be used.

Similarly, the term *collection* has been changed to *index* .

We recommend that you start using the newer variables and parameters so as to avoid confusion.

For more information about how to implement variables in a search or print template, or how to implement search parameters in an HTML document to create a search form, see Chapter 9, "Customizing Your Search Solutions," on page 77.

Global Template Variables

Global template variables are ones that can be used in any of the Web Search templates.

Table 4 Global Template Variables

Variable Name	Description
$$Authenticated	Indicates whether or not the user is authenticated by returning either a 1 or 0.
$$BeginAuthenticated	Begins a section for a valid, logged in user. Used in conjunction with $$EndAuthenticated. If a user is authenticated, the text between these two tags will be processed and appear in the output. If a user is not authenticated, the text will be removed from the search result. To control the appearance of unauthenticated search results, see $$BeginUnAuthenticated.
$$BeginLoop	End of the header section. Beginning of the repeating body section. This section is repeatedly parsed until there are no further result items to process. See also "$$EndLoop" on page 61.
$$BeginIndexesLoop	Begins a repetitive section that will be processed for each index the user specified in the search query. See also "$$EndIndexesLoop" on page 61 and "$$QueryIndex[number]" on page 62.
$$BeginFiltersLoop	Begins a repetitive section that will be processed for each filter parameter associated with the current query item. Note that multiple query items can be sent as part of a single query. See also "$$EndFiltersLoop" on page 61.
$$BeginQueryLoop	Begins a repetitive section that will be processed for each query item associated with the current search query. See "$$NumQueryItems" on page 62 for more information. See also "$$EndQueryLoop" on page 61.
$$BeginReturnFieldsLoop	The beginning of a repetitive section that will be reprocessed for each return field the user specified in the search query. See also "$$QueryReturnField[number]" on page 63.
$$BeginSortKeysLoop	Begins a repetitive section that will be processed for each sort key the user specified in the search query. See also "$$EndSortKeysLoop" on page 61 and "$$SortKeysCurrent" on page 64.
$$BeginUnAuthenticated	Begins a section for an unrecognized or logged out user. Used in conjunction with $$EndUnAuthenticated. If a user is not recognized, the text between these two tags will be processed and appear in the output. If the user is recognized as a valid, logged-in user, this text will not appear in the output. To control the appearance of authenticated search results, see $$BeginAuthenticated.

Variable Name	Description
$$Counter[*variable_number*, *increment_number*]	Inserts the value of the specified var# counter into the search result page. All counters initialize to zero. The optional second parameter specifies the amount to increment or decrement the current value. A maximum of 10 counters is supported. For example: $$Counter[1] = insert value of counter #1 $$Counter[1,1] = increment counter #1 by 1 and display the new value $$Counter[5,-3] = decrement counter #5 by 3 and display the new value
$$DefaultQueryEncoding	Default encoding of user query if not specified using the &encoding= query parameter.
$$EndAuthenticated	Ends a section for a valid, logged in user. For more information, see "$$BeginAuthenticated" on page 60.
$$EndFiltersLoop	Ends a repetitive section that will be processed for each filter parameter associated with the current query item. See also "$$BeginFiltersLoop" on page 60.
$$EndIndexesLoop	Ends a repetitive section that will be processed for each index the user specified in the search query. See also "$$BeginIndexesLoop" on page 60 and "$$QueryIndex[number]" on page 62.
$$EndLoop	End of the repeating body section. Beginning of the footer section.
$$EndQueryLoop	Ends a repetitive section that will be processed for each query item associated with the current search query. See "$$NumQueryItems" on page 62 for more information. See also "$$BeginQueryLoop" on page 60.
$$EndReturnFieldsLoop	The end of a repetitive section that will be reprocessed for each return field the user specified in the search query. See also "$$QueryReturnField[number]" on page 63.
$$EndSortKeysLoop	Ends a repetitive section that will be processed for each sort key the user specified in the search query. See also "$$BeginSortKeysLoop" on page 60 and "$$SortKeysCurrent" on page 64.
$$EndUnAuthenticated	Ends a section for an unrecognized or logged out user. For more information, see the description for the $$EndUnAuthenticated variable.
$$FilterCount	Identifies the number of filters associated with the current query item. Note that multiple query items can be associated with a single query.
$$FilterCurrent	Identifies the number of the current filter associated with the current query item.
$$FilterOperator	Identifies the boolean operator used to join the filters associated with the current query item. The full set of filters is always joined to the current query item using the Boolean AND.

Variable Name	Description
$$FilterValue[number]	Pulls the value of the specified filter associated with the current query item. If the optional # parameter is not provided, the current filter loop value ($$FilterCurrent) is used.
$$IncludeFile[*template_name*]	Automatically pulls in the designated template at the location of this variable. The included template can contain other template variables, which will be processed as though they were a part of the original template. The template name parameter can either be a full FILE:// URL based on the file system of the server or a relative path based on the location of the parent template. The template name parameter can be located within quotation marks. See the SearchResultTemplate.html file for an example use of this variable.
$$IndexesCount	The number of indexes associated with the user query. See also "$$BeginIndexesLoop" on page 60.
$$IndexesCurrent	The number of the current index. See also "$$BeginIndexesLoop" on page 60.
$$NumQueryItems	The number of query items contained within the current query. While most queries use only 1 query item, it is possible to construct a query with multiple search criteria, each weighted with a value between 1 and 100. While the resultant search contains hits from each of the queries, the search results are organized with the most relevant hits first (from any of the individual queries).
$$Query[number]	Query specified by the client into the search field. The optional number identifies the corresponding query item. The value of $$QueryCurrent is used if the optional number is not provided. See also "$$NumQueryItems" on page 62 for more information.
$$QueryCount	The number of query items associated with the search query. See also "$$NumQueryItems" on page 62 for more information.
$$QueryCountry	The country requested by the client. Note that this must be an uppercase, two-character value as specified in ISO 3166-1.
$$QueryCurrent	The number of the current query item. See "$$NumQueryItems" on page 62 for more information. See also "$$BeginQueryLoop" on page 60.
$$QueryDate	The begin date requested by the client. Only those documents dated on or after the specified date will be returned in the search results. See the query parameter "date" on page 72 for more information.
$$QueryEncoding	Actual encoding used to interpret the query (this could be either the same as the $$DefaultQueryEncoding, the value of the &encoding= query parameter, or UTF8).
$$QueryFileFilter	Returns the filename filter associated with the user query.
$$QueryIndex[number]	The names of the indexes the user specified in the search query. If the optional number is not provided, the current value of $$IndexesCurrent is used. See also "$$BeginIndexesLoop" on page 60.
$$QueryLanguage	The language requested by the client. Note that this must be a lowercase, two-character value as specified in ISO 639.

Variable Name	Description
$$QueryNumHits	The number of search results requested by the client.
$$QueryOperator	The type of the current search: 0 = Boolean AND search 1 = Boolean OR search 2 = phrase search
$$QueryReturnField[number]	The name of the return fields the user specified in the search query. If the optional number is not provided, the current value of $$ReturnFieldsCurrent is used. See also "$$BeginReturnFieldsLoop" on page 60 and "$$EndReturnFieldsLoop" on page 61.
$$QueryServerName[text]	Identifies the name of the Virtual Search Server provided with the &server= query parameter. The optional text parameter can be provided in the following formats: $$QueryServerName = NameOfServer $$QueryServerName[text] = text NameOfServer $$QueryServerName[%text] = text URLEncodedNameOfServer $$QueryServerName[text $$QueryServerName text] = text NameOfServer text $$QueryServerName[%text $$QueryServerName text] = text URLEncodedNameOfServer text See also "$$ServerName" on page 64.
$$QueryTemplate	The template name requested by the client. See also "$$TemplateName" on page 64.
$$QueryTemplateTheme	The template theme requested by the client. This is not necessarily the theme of the search result since the specified theme may not exist. See also "$$TemplateTheme" on page 64.
$$QueryVersion	The version number of the current query format.
$$QueryWeight[number]	The weighting of the current query item 1-100. See "$$NumQueryItems" on page 62 for more information.
$$ResultEncoding	Identifies the encoding used to return the current search results page. This is either the value of the valid &retencoding= query parameter or the default specified by the search administrator in the NetWare Web Search Manager.
$$ReturnField[number]	The name of the return fields the user specified in the search query. If the optional number is not provided, the value of $$ReturnFieldsCurrent will be used. See also "$$BeginReturnFieldsLoop" on page 60 and "$$EndReturnFieldsLoop" on page 61.
$$ReturnFieldsCount	The number return fields specified in the search query. See also "$$BeginReturnFieldsLoop" on page 60.
$$ReturnFieldsCurrent	The number of the current iteration of the $$BeginReturnFieldsLoop.

Variable Name	Description
$$SearchFor[number]	Query entered by the client into the search field. If the optional number is not provided, the value of $$QueryCurrent will be used. See "$$Query[number]" on page 62 for more information.
$$ServerName	The name of the virtual search server that produced the current output. See also "$$QueryServerName[text]" on page 63.
$$ServerLocation	Path on the network server to the virtual search server configuration files and indexes.
$$SortField[*number*]	The name of the field to sort on. If the optional number is not provided, the value of $$SortKeysCurrent will be used. See "$$SortByURL[sortfield.sortorder ...]" on page 67 and the query parameter "sortfieldnumber" on page 76 for more information.
$$SortKeysCount	The number of sort keys associated with the current query.
$$SortKeysCurrent	The current sort keys number. See "$$BeginSortKeysLoop" on page 60 for more information.
$$SortOrder[number]	The order to sort the field (ascending, descending, and default). If the optional number is not provided, the value of $$SortKeysCurrent will be used. See "$$SortOrder[number]" on page 64 for more Information.
$$TemplateLocale	Identifies the locale of the template, such as zh_TW. The locale information is taken from the template filename.
$$TemplateName	Identifies the filename of the template currently displayed in the browser. See also "$$QueryTemplate" on page 63.
$$TemplateTheme	Identifies the theme (or theme directory) that the current template belongs to. See also "$$QueryTemplateTheme" on page 63.

Search Page Variables

In addition to the global template variables, the following table lists all available search page variables that can be used to extend the functionality of the default search templates (SearchTemplate.html or SearchTemplate.Simple) or to create new templates from scratch.

Table 5 Search Page Variables

Variable Name	Description
$$BeginServerIndexesLoop	Begins a repeating section in the search template where information for each of the defined indexes will be written.
$$EndServerIndexesLoop	Ends a repeating section in the search template where information for each of the defined indexes will be written.
$$ServerIndexDescription	Inserts the description of the virtual search server defined in the Web Search Manager.
$$ServerIndexName	Inserts the name of the virtual search server defined in the Web Search Manager.

Search Result Variables

In addition to the Global Template Variables, the following table lists all available search result variables that can be used to extend the functionality of the default search result templates or to create new templates from scratch.

Table 6 Search Result Variables

Variable Name	Description
$$Author	The name of the original author of a document returned in a hit.
$$BeginAuthorized	Begins a section for a search result that the user has rights to see. Used in conjunction with $$EndAuthorized. If a search result is authorized, this section of text and template variables will be processed. If unauthorized, this section is removed from the output. See also "$$BeginUnAuthorized" on page 65.
$$BeginUnAuthorized	Begins a section for a search result that the user does not have rights to see. Used in conjunction with $$EndUnAuthorized. If a search result is not authorized, this section of text and template variables will be processed. If the search result is authorized, this section is removed from the output. See also "$$BeginAuthorized" on page 65.
$$EndAuthorized	Ends a section for a search result that the user has rights to see. For more information, see the description for the $$BeginAuthorized variable.
$$EndUnAuthorized	Ends a section for a search result item that the user does not have rights to see. For more information, see the description for the $$EndUnAuthorized variable.
$$Index	The name of the index in which a particular search result item was found.
$$DateTime[*date_format*]	The date and time of a hit. This is automatically written in Java's medium format using the client's locale (all calendars, translations, date and time formats are observed). $$DateTime[ ] can use an optional date and time format provided within the brackets [ ]. The text should conform to the Java *DateFormat* syntax.
$$Description	The abstract, description, or first 255 display bytes of the result item.
$$FileFormat	Indicates a specific document type. For example, .DOC or .PDF.
$$FirstHit	The hit number of the first item in the current result page. Is displayed using the client's locale.
$$Language	The language of the result item. $$Language is displayed in the language of the client's locale.
$$LastHit	The hit number of the last item in the current result page. Is displayed using the client's locale.

Variable Name	Description
$$MoreHits[*page#*, *text*]	A conditional text section to be included only if there are additional hits in the search results that can be retrieved. If the first section of the conditional text contains a number followed by a comma (for example, $$MoreHits[3, text to be included]), then the server will first determine if the designated search results page exists. If the page# is missing, 1 (the next page) is assumed. If the designated page is available, the remaining text after the comma and up to the closing bracket is written to the result page. Note that the initial number is relative to the current page. That is, -1 references the page immediately before the current page and 1 references the page immediately after. Zero refers to either the previous page or the next page.
$$MoreHitsURL[*page_number*]	The URL needed to display another page of search results. The optional parameter identifies the desired search result page number. If not provided, 1 is assumed. Note that the page number is relative to the current page. That is, -1 refers to the page immediately before the current page and 1 references the page immediately after. Zero (0) refers to the current page. The URL is inserted only if the designated page exists.
$$Number	The hit number of the current result item. Possible numbers begin with 1 and end with $$TotalHits. Is displayed using the client's locale.
$$PageNum[*page#*]	Inserts a user-specified search result page number. The optional page# parameter identifies the relative page from the current result page. That is, minus one (-1) refers to the page immediately before the current page and one (1) references the page immediately after. Zero (0) refers to the current page. The page number is inserted only if the designated page exists.
$$PrintURL[*first_hit_number*, *number_of_hits*]	The URL used to print the hits listed on the current search result page. The optional parameters can be specified to define the beginning search result number and the number of search results to include in the print job. The *number_of_hits* parameter can use the $$TotalHits template variable.
$$Relevance	How closely the result matches the user's query, indicated by percentages (1% to 100%).
$$SearchFor[number]	Query entered by the client into the search field. See "$$Query[number]" on page 62 for more information.
$$SearchTime	Inserts the amount of time used to process the current search request. $$SearchTime is displayed using the client's locale.
$$Size	The size of the data pointed to by the result item's URL. Is displayed using the client's locale.

Variable Name	Description
$$SortByURL[*sortfield.sortorder* ...]	The URL used to show the current result page sorted by one or more search result fields. Sort field names include title, author, changedate, filelength, language, summary, relevance, url, index, format, and document_number. Optional sort orders include ascending and descending. Sort field and sort order names are separated by a period. Multiple sort fields are separated by a space.
$$Title	If a title is not available in documents being searched, $$URL is used instead; if the URL is unavailable, < *title unavailable* > continues to be used.
$$TotalHits	The total number of hits that match the search query. This is not the same as the number of hits displayed in any particular result page. Is displayed using the client's locale.
$$URL	URL of the result item.

Print Result Variables

In addition to the Global Template Variables, the following table lists all available print result variables that can be used to extend the functionality of the default print result templates or to create new templates from scratch. For more information about how to implement variables in a template (HTML) page, see Chapter 9, "Customizing Your Search Solutions," on page 77.

Table 7 Print Result Variables

Variable Name	Description
$$BeginAuthorized	Begins a section for a search result that the user has rights to see. Used in conjunction with $$EndAuthorized. If a search result is authorized, this section of text and template variables will be processed. If unauthorized, this section is removed from the output. See also "$$BeginUnAuthorized" on page 65.
$$BeginUnAuthorized	Begins a section for a search result that the user does not have rights to see. Used in conjunction with $$EndUnAuthorized. If a search result is not authorized, this section of text and template variables will be processed. If the search result is authorized, this section is removed from the output. See also "$$BeginAuthorized" on page 65.
$$EndAuthorized	Ends a section for a search result that the user has rights to see. For more information, see "$$BeginAuthorized" on page 65.
$$EndUnAuthorized	Ends a section for a search result item that the user does not have rights to see. For more information, see "$$BeginUnAuthorized" on page 65.

Variable Name	Description
$$BeginTOCList[*text*]	Beginning of the table of contents repeating section. This section is repeatedly parsed until there are no further TOC result items to process. This is a conditional text section. The items within the brackets ([ ]) are processed only if the current item represents a change in the depth of the hierarchy. If $$Product appears within the conditional text, it will be replaced only if the current item also represents a new product.
$$Bookmark	The HTML anchor name of the current result item. This can be used to jump from a TOC entry to the corresponding section within the print job. All bookmark entries begin with “novell_print_toc_” and are followed by the number of the current result item, as in novell_print_toc_1.
$$Description	The abstract, description, or first 255 display bytes of the result item.
$$EndTOCList[*text*]	End of the table of contents section. This is a conditional text section. The items within the brackets ([ ]) are written out each time a result item occurs that decreases the depth of the hierarchy. If the depth of the current item is several levels less than the previous item, the text within the conditional text block is written out that many times.
$$Number	The hit number of the current result item. Possible numbers begin with 1 and end with $$TotalHits. Is displayed using the client’s locale.
$$NumIndents	The number of indentations required for the current Table of Contents entry.
$$Product	The name of the product associated with the current item in the table of contents. This displays only if this is the first result item within that product. See also “$$BeginTOCList[text]” on page 68.
$$Title	Title of the result item. For empty titles, <title unavailable> is displayed. Is localized using the client’s locale.
$$TotalHits	The total number of hits that match the search query. This is not the same as the number of hits displayed in any particular result page. Is displayed using the client’s locale.
$$URL	URL of the result item.
$$URLContent	The entire contents of the URL are placed into the template at this location. The URL contents are not parsed to validate their data type, formatting, or functionality. Only text/plain and text/html files are printed. All other files are inserted into the print job as an error message.

Error Message Variables

In addition to the Global Template Variables, the following table lists all available error message variables that can be used to enhance the organization of the default error message template, or to create new templates from scratch. For more information about how to implement variables in a template (HTML), see Chapter 9, “Customizing Your Search Solutions,” on page 77.

Table 8 Error Message Variables

Variable Name	Description
$$ErrorNumber	A numeric version of the error.
$$ErrorMessage	A text version of the error. Generally quite terse.
$$ErrorDescription	A longer version of the message. This might include additional error details or problem resolution information.

Response Message Variables

In addition to the Global Template Variables, the following table lists all available response message variables that can be used to enhance the organization of the default response message templates or to create new templates from scratch. For more information about how to implement variables in a template (HTML), see Chapter 9, “Customizing Your Search Solutions,” on page 77.

HINT: The repeating variables $$BeginLoop and $$EndLoop should not be used in a response message and will be ignored if used.

Table 9 Response Message Variables

Variable Name	Description
$$Cancel[*text*]	If the Cancel button is specified by Server logic, parses and inserts the conditional text into the response page.Currently used by PrintServlet when a print job exceeds the print job size warning limit.
$$Continue[*text*]	If the Continue button is specified by Server logic, parses and inserts the conditional text into the response page.Currently used by PrintServlet when a print job exceeds the print job size warning limit.
$$Help[*text*]	If the Next button is specified by Server logic, parses and inserts the conditional text into the response page.
$$Ignore[*text*]	If the Ignore button is specified by Server logic, parses and inserts the conditional text into the response page.
$$Next[*text*]	Compare to $$Prev.
$$No[*text*]	If the No button is specified by Server logic, parses and inserts the conditional text into the response page.
$$OK[*text*]	If the OK button is specified by Server logic, parses and inserts the conditional text into the response page.Currently used by PrintServlet when a print job exceeds the maximum print job size.
$$Prev[*text*]	If the Previous button is specified by Server logic, parses and inserts the conditional text into the response page.
$$ResponseNumber	A numeric version of the response required of the user.
$$ResponseMessage	A text version of the response required of the user. Generally quite terse. Can often be used as a title.
$$ResponseDescription	A longer version of the message. This might include additional details or *see also* type information.

Variable Name	Description
$$Retry[*text*]	If the Retry button is specified by Server logic, parses and inserts the conditional text into the response page.
$$URL	Inserts the URL to use when the parent button is clicked. This must appear within the brackets of a button's conditional text section.The URL logic is generated by the server.
$$Yes[*text*]	If the Yes button is specified by Server logic, parses and inserts the conditional text into the response page.

Search Parameters

The following table lists all available search parameters, including required syntax, a description of their default values, and examples. Each of these parameters can be used to extend or enhance the functionality of the search page templates or to create new search page templates from scratch. For more information about how to implement parameters in an HTML document, see Chapter 9, "Customizing Your Search Solutions," on page 77.

HINT: If you use a parameter but leave its value blank, the default value for that parameter will be used.

Table 10 Search Parameters

Parameter Name	Value	Description
server	String	Specifies the name of the virtual search server which is to receive this request. This query parameter is optional if the domain name of the request matches the name or alias of a registered virtual search server. **Syntax**: server=*virtual_search_server_name* **Example**: server=digitalairlines **Default**: domain name portion of search request
query*number*	String	The actual search criteria that is passed to the Web Search Server. The next six parameters below are combined with this parameter and are identified by adding the unique number to them. **Syntax**: query*number*=searchcriteria **Example**: query0=novell+AND+groupwise

Parameter Name	Value	Description
filternumber	String	The &filter#= query parameter is used to send additional query details not specified by the user to help limit the scope of a search. Normally, these would be included as hidden fields on an HTML form. This parameter supports all of the same features and functionality as the &query= parameter. However, unlike the &query= parameter, this parameter can be sent multiple times for a single query item. The individual filters associated with a single query item are joined using the value of the filteroperator parameter. The set of filters is logically joined with the rest of the query item using the boolean AND operator. **Syntax**: filter*number*=searchcriteria **Example**: filter0=/product=Groupwise See also "filteroperatornumber" on page 71.
filteroperatornumber	Number	This parameter identifies the boolean conjunction to be used between multiple filters (several filters can be associated with a single query item). The complete set of filters is always associated with the corresponding query item using the boolean AND operator. **0** = AND **1** = OR **2** = PHRASE **Syntax**: filteroperatornumber=*number* **Example**: filteroperator0=1 **Default**: None See also the query parameters "operatornumber" on page 72 and "filternumber" on page 71.
id*number*	String	A document ID that is used to narrow a search. You can specify more than one ID by using the same field name more than once. **Syntax**: id*number*=*documentID* **Example**: &id0=z1.0010.&id0=z1.0020 **Default**: None

Parameter Name	Value	Description
operator*number*	Integer	Indicates which operator to use between two or more words in a search. **0** = AND **1** = OR **2** = PHRASE **Syntax**: operator*number*=*number* **Example**: operator0=1
weight*number*	Integer	Lets you assign a level of importance to the current query item as far as it relates to the other query items that are part of the search query. Web Search Server uses this number along with the relevance number to determine a search results' final relevance and then orders the results accordingly. **Range**: 0 to 100 **Syntax**: weight*number*=*number* **Example**: weight0=75
type*number*	Integer	Indicates the type of search. Options include: **0** = Normal search; 0 is the default. **1** = Searches only the given document numbers. **2** = Root search used by the search tree control to get the top tree nodes. **3** = Used to get the children of the given document number. **4** = Searches the descendants of the given document numbers and is used to narrow a search or a print request, including all of its children. **Syntax**: type*number*=*number* **Example**: type0=2 **Default**: 0 (zero)
date	Integer	Lets you specify a date range to be searched in milliseconds. The example shows the number of milliseconds spanning a three-month time frame. The minus sign (-) before the number indicates three months back in time. If you pass a positive number such as 940457147873, then Web Search creates a date and time based on the number of milliseconds elapsed since January 1, 1970; 12:00 a.m. The example number 940457147873 produces the search start date of October 20, 1999, at 4:05:47 p.m. **Syntax**: date=*number* **Example**: date=-7905600000

Parameter Name	Value	Description
filefilter	String	Use this query parameter to filter search results based on their path, domain, filename, or extension. This parameter uses the same query syntax as the &query= parameter. **Syntax**: filefilter=*value* **Example**: filefilter=www.novell.com **Example**: filefilter=pdf **Default**: None
index	String	Lets you restrict a search to one or more specified indexes. The index name you specify using this parameter must exactly match the name of an index defined at the server. You can specify more than one index by either sending this parameter more than once or by separating the list of indexes with a semi-colon. **Syntax**: index=*index_name1[;index_name2]* **Example**: index=GroupWise&index=NetWare **Example**: index=GroupWise;NetWare
numhits	Integer	Indicates the number of hits you want returned at one time in the search results page. **Syntax**: numhits=*number* **Example**: numhits=25 **Default**: 25
starthit	Integer	Indicates the hit number you want Web Search to begin searching from. If you entered 35 as the STARTHIT parameter value, Web Search would return hits beginning with hit number 35. **Syntax**: starthit=*number* **Example**: starthit=35 **Default**: 1
lang	String	Lets you specify a language using the two-character, lowercase language value derived from ISO6391. **Syntax**: lang=*language_code* **Example**: lang=ja **Default**: browser language preference

Parameter Name	Value	Description
country	String	Lets you specify your country using the two-character, uppercase country value derived from ISO3166. **Syntax**: country=*country code* **Example**: country=TW **Default**: browser language preference
template	String	Lets you specify the specific results template you want your search results returned in. The following list of templates are the default templates included with the Web Search Server. However, your Web Search Server administrator might have created custom templates using different names. Check with your administrator if these templates do not work for you. You must type the names of these templates *exactly* as they appear in this list: ◆ ResultListTemplate.html ◆ ResultListTerseTemplate.html ◆ ResultListVerbose.html ◆ PrintResultTemplate.html Localized versions for multiple languages can also be used. See "Working with Multiple Languages" on page 81. **Syntax**: template=*filename* **Example**: template=ResultList.html
theme	String	The name of the theme, or directory within the templates directory, where a complete set of search and print templates are stored. **Syntax**: theme=*theme_name* **Example**: theme=Intranet
showfirsthit	Boolean	If true, rather than displaying the search results page, this parameter automatically goes to the URL of the first hit on the current page. **Syntax**: showfirsthit=*value* **Example**: showfirsthit=True **Default**: False

Parameter Name	Value	Description
retfield	String	Lets you determine the level of detail given about each result item. The fewer the details, the faster a search is returned to a user. Field names include title, author, URL, changedate, language, summary, relevance, index, format, and filelength. **NOTE:** Type these fields exactly as they appear. To specify more than one field, use the RETFIELD parameter and value, separated by an ampersand (&) as in the following: retfield=title&retfield=author **Syntax**: retfield=*field_name* **Example**: retfield=title
buttonpressed	String	A button pressed by the user. If this value is part of the query, then a response message should not be sent to the client. Options include Yes, No, OK, Cancel, Continue, Ignore, Retry, Prev, Next, and Help. **Syntax**: buttonpressed=*button_name* **Example**: buttonpressed=Cancel
gettotalhits	Boolean	Lets you enable or disable the total number of hits calculation. For example, if you set the GETTOTALHITS parameter to FALSE, the Total Number of Hits label on the results page will display 0 (zero). Setting this parameter to TRUE will show the total number of hits found during the search. In some complex situations this can save valuable processing time. **Syntax**: gettotalhits=*value* **Example**: gettotalhits=false **Default**: True
encoding	String	Specifies the character set encoding used to encode the search request itself. **Syntax**: encoding=*value* **Example**: encoding=Shift_JIS **Default**: UTF-8
retencoding	String	Lets you specify the character set encoding to be used by the next results page returned to the user. **Syntax**: retencoding=*character_set_encoding* **Example**: retencoding=iso-8859-1 **Default**: UTF8

Parameter Name	Value	Description
sortbydate	String	Sorts the Total Search Results list by date, ignoring the normal relevance ordering. **Syntax**: sortbydate=*value* **Example**: sortbydate=true **Default**: True
relevance	String	Tells Web Search whether or not to sort the search results by relevance. Turning this feature off is a potential speed gain since the sort algorithm will then not run. **Syntax**: relevance=*value* **Example**: relevance=false **Default**: true
sortkeys	Integer	Lets you specify the number of sort fields that should be used to sort the search results. **Syntax**: sortkeys=*number* **Example**: sortkeys=1
sortfieldnumber	String	Allows you to specify the fields on which to sort the search results returned in a results page. Grouped together with the sortorder query parameter by adding a *number* to the end of the parameter name. Field names include title, author, URL, changedate, language, summary, relevance, index, format, and filelength. **IMPORTANT:** Type these fields *exactly* as they appear above. **Syntax**: sortfield*number*=*field_name* **Example**: sortfield1=title
sortordernumber	Integer	Lets you specify the alphanumeric ordering of search result items (hits). Grouped together with the sortfield query parameter by adding a *number* to the end of the parameter name. Options include the following: **0**= Ascending **1**= Descending **2**= Default for each field. **Syntax**: sortorder*number*=*number* **Example**: sortorder1=0

9 Customizing Your Search Solutions

You can quickly create a custom search solution by modifying the default NetWare® Web Search templates. Templates include some fundamental options for users, but you can add or remove options and modify the form layout and design to give the search form the look, feel, and function you need. If you are creating a hosted search service for another company's Web site, you can modify the templates to match the look and feel of their Web site.

If you are confident in coding with HTML, you can start with the default search page template to get a feel for the available parameters and then begin coding completely new search and print templates from scratch.

For more information about the necessary components for building a solution from scratch, see Chapter 6, "Understanding Templates," on page 43 and Chapter 8, "Working with Template Variables and Search Parameters," on page 59.

Customizing Templates

You can extend the capabilities of NetWare Web Search Server by customizing the templates.

The first step is to determine which components of Web Search you want to customize. For example, if you only want to add a few additional search features to the search page template and modify its background color and table size, you would modify the SearchTemplate.html or SearchTemplate.Simple files.

This section discusses how to customize the search, print, and result templates and how to use available parameters and variables to create a customized search solution.

Customizing the Search Templates

You can customize the design and functionality of the static or dynamic search templates by

- Modifying HTML code
- Adding or removing search parameters

If you are familiar with HTML, you can quickly modify the design of the default (dynamic) search template or the static search template. For example, you can change the colors of the search page or add new custom graphics.

To modify the functionality of the default search template, you can add or remove search parameters. Search parameters are used to communicate with NetWare Web Search. By embedding them in the correct places in your HTML source, you can extend or limit the functionality of the default search template.

For example, if you wanted your users to use a specific set of templates found in a themes directory, you would add the following HTML code, including the theme parameter, to the SearchTemplate.html file:

```
<INPUT TYPE="Hidden" NAME="theme" VALUE="$$QueryTemplateTheme">
```

This sample HTML code tells Web Search to look for templates only in the specified template directory. All themes are located within the templates directory specified in Web Search Manager.

For a complete list of available search parameters, see Table 10, “Search Parameters,” on page 70.

Customizing Search Result Templates

NetWare Web Search Server includes several default search result templates that are used to display hits, provide feedback to a user, or request information from a user after a search is performed. For more information about the default search result templates, see Chapter 6, “Understanding Templates,” on page 43.

You can customize the design and functionality of the default search result template, the template used when a user selects Normal from the Result List Format drop-down list in the NetWare Web Search form. For information about how to access the NetWare Web Search form, see “Taking a Test Run” on page 11.

Customizing the default search result template involves

- Modifying the HTML code
- Adding or removing search result variables

If you are familiar with HTML, you can quickly modify the design of the default search result template. For example, you can change the colors of the search page or add new graphics.

To modify the functionality of the default search result template, you can add or remove search result variables. Search result variables are placed in the template where you want search results to be displayed.

For example, if you wanted to display the total number of hits returned when a user performs a search and you wanted the information to appear in the upper-left corner of the search results page, you would add the following HTML code to the search result template file:

```
Total Search Results: $$TotalHits
```

After a user performs a search, the $$TotalHits variable would be replaced by the actual total number of hits found during the search.

The $$TotalHits variable is used to retrieve the total number of hits found during a search. You can place this variable anywhere in the results list template to organize the display of information in the way you want.

Default search result templates are located in *volume*:*searchroot*\TEMPLATES\. For a complete list of search result variables that you can use to customize default search result templates or to create new ones, see Table 6, “Search Result Variables,” on page 65.

Customizing Print Result Templates

NetWare Web Search Server includes two default print result templates: the default print result template and a "no hits" template. Print result templates are used to organize and format search results for printing and to provide feedback to a user when no hits are found. For more information about the default print result templates, see Chapter 6, “Understanding Templates,” on page 43.

You can customize the design and functionality of the default print result template in the same way you customize the search result template by

- Modifying the HTML code
- Adding or removing print result variables

If you are familiar with HTML, you can quickly modify the design of the default print result template. For example, you can change the colors of the print results page or add new graphics to it.

To modify the functionality of the default print results template, you can add or remove print result variables. Variables are placed in the template where you want search results to be displayed.

For example, if you wanted to remove the table of contents from the default print results template, you would remove, or comment out, the following HTML code in the PRINTRESULTLIST.HTML template, which would include the $$BeginTOCList variable:

```
<CENTER><H2>Table of Contents</H2></CENTER><p><!--    TABLE OF CONTENTS
-->$$BeginTOCList[<BIG><B>$$Product</B></BIG><DL>]<DT><A
HREF="#$$Bookmark"><BIG>$$Title</BIG></A><SPACER TYPE=HORIZONTAL
SIZE=20><I><SMALL>[$$URL]</SMALL></I>$$EndTOCList[</DL>]
```

You could either save your changes in the default print result list template or you could save it using a new name, thereby creating an alternative template for users who don’t want a table of contents in the print results. To be effective, you would then have to add a hypertext link in the search result template that would include the &template=*new_template_name* query parameter.

Default print result templates can be found at *volume*:\searchroot\TEMPLATES. For a complete list of print result variables that you can use to customize default search result templates or to create new ones, see Table 7, “Print Result Variables,” on page 67.

Customizing Error and Response Message Templates

Error and response messages are used to either provide feedback to the user or to request information from the user.

Error and response message templates are used to display the content of error and response messages sent by the Web Search Server in response to search or print errors. Similar to search and print templates, error and response templates can be customized. However, because the contents of error and response messages are built into NetWare Web Search Server, you cannot modify the contents of the messages or the button objects that might appear, depending on the type of response being generated.

Customizing Error Messages

There are several error messages that can be returned to a user. For example, when users incorrectly use a search operator in a search form, they might get the message, "Search Error: Incorrect use of Boolean operator." An error number might also appear.

While you can utilize HTML tags to format an error message, add or remove variables to determine what information is shown to the user, or even reorganize where the messages will appear in the template, you cannot modify the message itself.

Customizing Response Messages

The same concepts apply to response messages, but response messages return buttons that a user can click. Which buttons appear are determined by the NetWare Web Search Server. While you can modify the labels of these buttons, you cannot determine which buttons will appear, or when.

Testing Your Search and Print Solution

Once you've completed customizing the templates and the search form, you can test them in your Web browser by pointing to the search form URL and entering a search string. See "Taking a Test Run" on page 11 for information about how to access the NetWare Web Search form.

HINT: Remember that a search cannot be performed until you have defined at least one index and generated it using NetWare Web Search Manager. Refer to Web Search Manger's online Help for the steps required in defining and generating an index. Also, see Chapter 5, "Creating and Managing Virtual Search Servers," on page 29 for an overview of indexes.

10 Internationalizing Your Search Solution

NetWare® Web Search Server is capable of handling search queries, search results, templates, and Web content in many languages and character sets. Web Search can auto-detect languages and character sets, but to ensure a complete international search solution, you must identify language, country, and character information throughout your Web Search implementation.

This chapter discusses all of the issues related to supporting multiple languages from a single search solution.

Working with Multiple Languages

Customizing your search solution is important only if you want to let your users conduct language-specific searches. You specify the language of a template by inserting a language identifier in the META tag of your templates or HTML files. The language identifier can also be used in Search Results pages to let users quickly recognize the search results that interest them.

NetWare Web Search Server also lets Web clients specify their locale at the time the search query is entered. The default Search page illustrates this feature by auto-detecting a user's locale and selecting the appropriate language from the Display Language drop-down list. This selection sends two parameters to the Web Search Server: language and country. The country parameter is almost always blank. The search engine uses this information to find locale-specific versions of the templates used to return search results.

To specify the language of a template or of any HTML content that gets indexed as part of your virtual search server, you must enter a language identifier within an HTML file's header section. For example, if you wanted to identify a Russian template, you would add the following META tag:

```
<meta http-equiv="Content-Language" content="ru">
```

In some cases, such as Traditional and Simplified Chinese, you will need to use the two-character, uppercase country codes. For example:

```
<meta http-equiv="Content-Language" content="zh-TW"><meta http-
equiv="Content-Language" content="zh-CN">
```

The first line of the example indicates the Chinese language (ZH) and the geographic location as Taiwan. The second line of the example indicates the Chinese language (ZH) but China as the geographic location.

This combination of language and country codes is called a *locale*. For more information about locales, refer to Table 11 on page 87.

Specifying Locales within Template Filenames

NetWare Web Search Server consists of three primary servlets: SearchServlet, PrintServlet, and AdminServlet. Each servlet returns information to the Web client using server-side templates. Templates are stored on at *volume*:\searchroot\TEMPLATES. For more information about templates, see Chapter 6, "Understanding Templates," on page 43.

After determining a Web client's locale, Web Search attempts to locate a matching search result template. That is, each of the Web Search services automatically attempts to locate a version of the requested template that most closely matches the Web client's locale.

IMPORTANT: NetWare Web Search cannot find locale-specific templates without the two-character language code and the optional two-character country code. See Table 11 on page 87 for more information about language code syntax.

For example, if a Web client requests to see search results using the ResultListTemplate.html file and the client is a Chinese language user from Taiwan and the server is Russian, then Web Search will try to find a Chinese-Taiwan version of the template first (ResultListTemplate_zh_TW.html) because that exactly matches the client's language and country. The following table lists the template names the system would look up in this example in order of priority.

Template Name	What Web Search Concludes
1. ResultListTemplate_zh_TW.html	Specific client locale
2, ResultListTemplate_zh.html	Simplified client locale
3. ResultListTemplate.html	Client requested name
4. ResultListTemplate_ru.html	Specific server locale (no simplified versions)
5. ResultListTemplate_en.html	English language version
6. ResultListTemplate.html	Up to the first underscore (_)

If this scenario were reversed so that the search client was Russian and the server was Chinese (Taiwan), and the client requested the ResultListTemplate_ja.html template, then the lookup order would follow the order shown in the following table.

Template Name	What Web Search Concludes
1. ResultListTemplate_ja_ru.html	Specific client locale (no simplified versions)
2. ResultListTemplate_ja.html	Client requested name
3. ResultListTemplate_ja_zh_TW.html	Specific server locale
4. ResultListTemplate_ja_zh.html	Simplified server locale
5. ResultListTemplate_ja_en.html	English language version
6. ResultListTemplate.html	Up to the first underscore (_)

All templates undergo this rigorous lookup system. Once a template is located, its name is stored and associated with the original client locale so that all subsequent requests for that template from the same locale automatically find the template without performing the same rigorous lookup.

No further lookups are attempted for that combination of client locale and template name until the NetWare Web Search Server is restarted. If all template lookups fail, then an error message is returned to the client performing the search.

Understanding Character Set Encodings

A character set is a grouping of alphabetic, numeric, and other characters that have some relationship in common. For example, the standard ASCII character set includes letters, numbers, symbols, and control codes that make up the ASCII coding scheme.A *character set encoding* is the mapping of a character set to a value that can be understood and processed by a computer.

NetWare Web Search relies on character set encodings to identify the characters used when performing a search, reading a template, posting results to a Web browser, or indexing Web-based content. If the encoding information is missing in any of these areas, NetWare Web Search uses the default encodings identified in the SearchServlet and PrintServlet properties files. You can modify these settings using NetWare Web Search Manager.

Because most languages have several encodings that their character sets are identified by, NetWare Web Search Server supports a wide variety of character set encodings and encoding aliases.

Some examples of character set encodings include iso-8859-1, shift_jis, big5, and latin2. The official list of registered encodings is available from the Internet Assigned Numbers Authority (see Table 11 on page 87). These are the official names for character sets that can be used in the Internet and can be referred to in Internet documentation. However, not all IANA-registered character set encodings are supported by NetWare Web Search Server. Refer to Table 11 on page 87 for a list of encodings and encoding aliases that are supported by NetWare Web Search Server.

Unicode and UTF8

Unicode is a 16-bit character encoding standard developed by the Unicode Consortium. By using two bytes to represent each character, Unicode enables almost all of the written languages of the world to be represented using a single character set. Unicode does not require any special processing to access any character in any language.

This makes Unicode very easy to use when processing text from multiple languages and scripts. This is the reason NetWare Web Search converts all external files into Unicode for processing.

As already mentioned, Unicode is two bytes wide for all characters. Although this is ideal for computer processing, it doubles the size of all single-byte languages. This has a significant impact on Internet performance. For this reason, NetWare Web Search also supports an alternate representation of Unicode known as UTF-8. UTF-8 is a Unicode Transformation Format that uses sequences of 1 to 6 bytes to represent all the characters in the Unicode standard. Most notably, ASCII characters are transmitted without any conversion at all. This means that most Internet content is already in the UTF-8 representation. Many Asian languages, however, require three bytes per character in the UTF-8 format. Other languages can require up to six bytes to represent each of their characters.

You will have to decide if Unicode or UTF-8 best meets your needs when creating HTML content, Web Search templates, or search pages.

Search Encodings

The only encodings NetWare Web Search currently supports when performing a search are Unicode and UTF-8. Therefore, any page that allows Web users to enter a search must ensure that the results are passed to the server in one of these two formats. See "Template Encodings" on page 85 for more information.

To pass Unicode characters to NetWare Web Search, use the syntax %uHHHH, where

- Percent sign (%) is used as the CGI escape character
- Lowercase letter U (u) indicate that the subsequent 4 characters represent a Unicode value.
- Four uppercase H letters (HHHH) indicate four hexadecimal characters (0-9, A-F)

To pass UTF-8 characters to NetWare Web Search, just use normal ASCII characters or the syntax %HH... for all other characters, where

- % is the CGI escape character
- HH indicates two he xi decimal characters (0-9, A-F)
- %HH indicates additional %HH groupings that might be required to properly transmit a character

HINT: If the encoding of the page containing a search form is already set to UTF-8 or Unicode, most browsers automatically transmit the entered search text correctly using the designated encoding.

By default, NetWare Web Search uses UTF-8 in its sample search pages.

Response Encodings

One of the many parameters that can be sent when conducting a search is the encoding that should be used when returning the results back to the browser. All NetWare Web Search encodings listed in Appendix B, "Combined Character Sets for Use with NetWare Web Search," on page 91 can be used.

If the search result page contains the ability to refine or redo the search, then the response encoding can significantly impact the possible characters that can be entered when conducting the next search from this page. For example, if the user requests results in the iso-8859-1 encoding (HTML's default), then only iso-8859-1 characters can be entered in the subsequent search from that page. Other characters can still be sent to the Web Search services using the %uHHHH and %HH formats, but the browser will not allow users to enter normal text characters other than that supported by iso-8859-1.

Although Web Search can return search results from many languages, some characters found in titles and descriptions might be returned as question marks (?) indicating that these characters are not available in the current response encoding. If a character can be represented in the current encoding but a font is not available, many browsers will substitute an alternate character such as an empty box character. Once the appropriate fonts have been installed, these characters will then display properly.

By default, NetWare Web Search returns all search, print, and administration pages in UTF-8.

HTML Encodings

Since HTML content can contain text written in many character sets, all HTML files need to include a tag that identifies the character set encoding. To identify the encoding of an HTML file (or search template), use the following META tag at the top of the file's header section:

```
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
```

In this example, you would replace Shift_JIS with the appropriate Internet Assigned Numbers Authority (IANA)-assigned encoding value.

It is very important that the CHARSET value accurately represent the character set encoding that was actually used when the HTML Web content or Web Search template was created. A correct entry allows Web Search to accurately interpret and convert the characters in the document. An incorrect entry prevents Web Search from being able to read the characters as valid data in the authored language.

IMPORTANT: Improperly identified characters result in garbled text. In some cases, the Web-based content cannot be properly indexed or printed. In the most severe cases, the document being read might produce a server-side exception, which will ultimately discontinue processing the document and perhaps the entire current operation.

Because Web Search is Unicode-based, when reading templates or when indexing or printing HTML content, all character encodings are converted from their source encoding to Unicode for internal processing.

During indexing, if a document contains characters not supported by the designated encoding, if the document doesn't have an encoding designation, or if the designation is inaccurate, the indexer will do its best to recover. But if it cannot, it might index the information incorrectly or quit indexing that page entirely.

When reading a template file, Web Search might automatically cease processing the file if it contains any characters not supported by the current encoding. It will try to ignore the invalid text and continue, but this might not be possible.

When displaying search results or when printing HTML content, any character that does not match the specified response encoding will receive a question mark (?) in its place when rendered at the browser. Although some characters are properly supported by the current encoding, the browser might not have the required fonts to display the characters. In this case, users might see square boxes representing these characters. This is an indication that the valid character reached the browser, but the operating system could not provide a font to properly render the character. The user would than have to either change fonts or install the correct fonts in order to properly display the characters.

HINT: If a document does not contain a CHARSET encoding value, the default encoding for HTML documents is ISO-8859-1, also known as Latin1. The default encoding for plain text documents is US-ASCII.

Web Search also allows administrators to define the default encodings for templates, HTML content when printing, and search and print responses. Refer to the NetWare Web Manager Help for information about changing the default encodings.

Template Encodings

All HTML documents should include a Content-Type META tag identifying their character set encodings. The character set encoding allows HTML Web clients (or browsers) to understand the contents of the file. This tag is also used by browsers to automatically switch their display system and fonts to correctly show the Web page's contents. This lets users surf the World Wide Web without having to constantly change their display system as they encounter content from various languages and characters sets.

However, because NetWare Web Search lets administrator specify both template encodings and response encodings, browsers might get confused when presented with the valid response encoding in the HTTP header and one or more alternate encodings from the Content-Type META tags within the file that was part of the original Web Search template.

NOTE: $$IncludeFile[ ] templates can also contain their own Content-Type meta tags.

To solve this problem, NetWare Web Search allows placing the Content-Type META tag specifying the template's encoding within an HTML comment. This effectively obscures the original template encoding from the browser, but still allows Web Search to read the encoding when the template file is processed.

A sample Web Search template is illustrated below. The Content-Type META tag has been hidden inside of an HTML comment. This template can be embedded within other templates using the $$IncludeFile[ ] template variable without affecting Web Search's ability to distinguish between the various encodings. This file can also be processed and then sent to a user's Web browser without conflicting with the response encoding provided by Web Search in the HTTP response headers.

```
<html>
<head><!-- Note that the HTML encoding command (meta tag) is hidden within a
comment so that it does not affect a user's browser display. --><!-- The
actual encoding used when sending this file to the user is controlled by the
response encoding --><!-- <META HTTP-EQUIV="Content-Type" CONTENT="text/
html; charset=iso-8859-1"> --></head>
<body>

Template data here.</body>
</html>
```

Encoding Issues When Printing

When NetWare Web Search processes a print request, it gathers the entire contents of each file and builds an appended print job page, one file after another. Each file can contain its own Content-Type META tag identifying its encoding. Each file's encoding will be used by Web Search to convert that file into Unicode before being sent out using the response encoding.

Unfortunately, all of these encoding META tags might confuse the browser's display system. While Web Search has already properly converted the files into a single response encoding, the browser sees the Content-Type META tags which direct it to do something else, and gets confused.

The way to solve this problem is to create a print results template that contains a Content-Type META tag encoding at both the top and bottom of the file, before and after the various documents get printed. All current browsers take either the first Content-Type META tag that they encounter or the last. Constructing a print template with both satisfies all browsers.

Languages Included in the Default Templates

There are additional search and print templates for each of the following languages:

- Chinese (Traditional and Simplified)
- English
- French
- German
- Italian
- Japanese
- Korean
- Portuguese

- Russian
- Spanish

Templates are stored at *volume*:*searchroot*\TEMPLATES.

What's Next

The following table lists additional resources for learning more about locales, country and language codes, and encodings.

Table 11 Additional Information Resources

Component	Resource Location
Language and country codes (locale)	RFC1766 (http://www.ietf.org/rfc/rfc1766.txt) **NOTE:** While RFC1766 uses the hyphen character (-) to separate language and country information, Web Search uses the underscore character (_) in order to conform to the Java convention. ISO639 (http://www.ics.edu/pub/ietf/http/related/iso639.txt) ISO3166 (http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html)
Character sets	Internet Assigned Numbers Authority (IANA) Character Set registry (http://www.isi.edu/in-notes/iana/assignments/character-sets)
Unicode	Unicode Consortium home page (http://www.unicode.org)
UTF-8	"UTF-8: A Transformation Format of ISO10646" (ftp://nis.nsf.net/internet/documents/rfc/rfc2279.txt)

A Troubleshooting NetWare Web Search

This appendix provides some troubleshooting topics that can help you overcome search and print performance issues.

Troubleshooting

Characters of descriptions or titles appear as intelligible characters

Possible Cause: You've probably indexed documents written in multiple languages and encodings. Web Search can index most of the world's languages and encodings. However, Web Search needs to know the encoding of each document.

Possible Cause: Some of your documents were probably not tagged with an encoding or were incorrectly tagged.

Action: Make sure all of your documents contain the correct Content-Type META tag. If your international documents do not contain a Content-Type META tag, either add it or use the Encoding (If Not in META Tags) index definition option to specify the default encoding.

Several titles or descriptions contain the same text

Possible Cause: If search results include duplicate titles or descriptions, your description fields (description, summary, or abstract) might include boilerplate information.

Action: The more accurate your META tag description fields are, the better your search results will be. Where possible, consider adding descriptions to your document's META tags.

Possible Cause: It could also be that you have indexed the same document more than once, or several links throughout your Web site might point to the same document but do so using different character cases each time.

Action: To solve the latter problem, try using the URLs Are Case Sensitive option to direct Web Search to turn off case-sensitive crawling. Also, remove any duplicate backup files you might have and exclude any backup directories from your index definition.

Some titles are returned as the URL of the document instead

Possible Cause: Web Search pulls document titles from within each document that it indexes. If your document doesn't have a title, Web Search uses the URL or path of the document instead. If the URL is unavailable, a Title Unavailable message is returned.

Action: Make sure all of the documents you index have specifically defined titles.

Additional Assistance

If the problem you are working with doesn't appear in this appendix, visit the Novell® Support Connection Web site (http://support.novell.com).

B Combined Character Sets for Use with NetWare Web Search

The following tables list the character set encoding names and aliases that Web Search recognizes when indexing, searching, displaying, or printing files. This information is a subset of the character names registered by the Internet Assigned Numbers Authority (IANA).

Whenever possible, the items listed in the first column of each table are the preferred MIME names listed in the Internet Assigned Numbers Authority (IANA) Character Sets registry. If a preferred MIME name is not available, items in the first column represent the primary registered names.

Items in the second column of each table are aliases which are also at times used to identify that encoding.

Note that not all aliases exactly represent the parent encoding under which they are listed. In these cases, they overlap significantly enough that they will be handled identically by the various NetWare® Web Search engines.

HINT: Character encodings appear in the exact case specified in the Internet Assigned Numbers Authority (IANA) Character Sets registry. Some uses of these encodings are case sensitive. However, NetWare Web Search ignores the case of these encodings.

ASCII Character Set

Preferred MIME Name or Primary Registered Name	Encoding Names
US-ASCII (MIBenum: 3)*	ANSI_X3.4-1968
	ANSI_X3.4-1986
	ASCII
	ascii7
	iso_646-us
	ISO646-US
	ISO_646.irv:1991
	iso-ir-6
	646
	us
	IBM367
	cp367
	csASCII
IBM437 (MIBenum: 2011)	ibm-437
	cp437
	437
	csPC8CodePage437

* A MIBenum is a record number corresponding to an entry in IANA's Management Information Base.

Arabic Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-8859-6 (MIBenum: 9)	ISO_8859-6:1987
	ISO_8859-6
	iso8859-6
	iso8859_6
	8859_6
	IBM1089
	ibm-1089
	cp1089
	1089
	iso-ir-127
	ECMA-114
	ASMO-708
	arabic
	csISOLatinArabic
Windows-1256 (MIBenum: 2256)	cp1256
	win1256
	ms1256

Chinese (Simplified) Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
gb2312 (MIBenum: 2025)	csGB2312
gb_2312-80 (MIBenum: 57)	iso-ir-58
	chinese
	csISO58GB231280
	gb2312-80
	gb2312-1980
	gb-2312-80

Preferred MIME Name or Primary Registered Name	Encoding Aliases
gbk	GBK
	windows-936
	ms936
	cp936
	cp-936
euc-cn	EUC_CN
	euccn
	euc-gb

Chinese (Traditional) Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
big5 (MIBenum: 2026)	Big5
	windows-950
	win950
	ms950
	csBig5
IBM950 (MIBenum: ????)	ibm-950
	cp950
	cp-950
	950

Cyrillic Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-8859-5 (MIBenum: 8)	ISO_8859-5:1988
	ISO_8859-5
	iso8859-5
	iso8859_5
	8859-5
	iso-ir-144
	IBM915
	ibm-915
	cp915
	915
	cyrillic
	csISOLatinCyrillic
KOI8-R (MIBenum: 2084)	koi8_r
	koi8
	cp878
	cp-878
	csKOI8R
Windows-1251 (MIBenum: 2251)	win1251
	cp1251
	ms1251

European Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
Windows-1252 (MIBenum: 2252)	cp1252
	ms1252
	win1252
	ansi
	ansi-1252

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-8859-1 (MIBenum: 4)	ISO_8859-1:1987
	ISO_8859-1
	iso8859-1
	iso8859_1
	8859_1
	iso-ir-100
	IBM819
	ibm-819
	CP819
	819
	l1
	latin1
	csISOLatin1
ISO-8859-2 (MIBenum: 5)	ISO_8859-2:1987
	ISO_8859-2
	iso8859-2
	iso8859_2
	8859_2
	iso-ir-101
	IBM912
	ibm-912
	cp912
	912
	l2
	latin2
	csISOLatin2

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-8859-3 (MIBenum: 6)	ISO_8859-3:1988
	ISO_8859-3
	iso8859-3
	iso8859_3
	8859-3
	iso-ir-109
	IBM913
	ibm-913
	cp913
	913
	l3
	latin3
	csISOLatin3
ISO-8859-4 (MIBenum: 7)	ISO_8859-4:1988
	ISO_8859-4
	iso8859-4
	iso8859_4
	8859-4
	iso-ir-110
	IBM914
	ibm-914
	cp914
	914
	l4
	latin4
	csISOLatin4
Windows-1250 (MIBenum: 2250)	cp1250
	ms1250
	win1250
IBM850 (MIBenum: 2009) (UNICODE)	ibm-850
	cp850
	850
	csPC850Multilingual

Preferred MIME Name or Primary Registered Name	Encoding Aliases
IBM852 (MIBenum: 2010)	ibm-852 cp852 852 csPCp852
IBM860 (MIBenum: 2048)	ibm-860 cp860 860 csIBM860
IBM863 (MIBenum: 2050)	ibm-863 cp863 863 csIBM863
IBM865 (MIBenum: 2052)	ibm-865 cp865 865 csIBM865

Greek Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-8859-7 (MIBenum: 10)	ISO_8859-7:1987
	ISO_8859-7
	iso8859-7
	8859_7
	IBM813
	ibm-813
	cp813
	813
	iso-ir-126
	ELOT_928
	ECMA-118
	greek
	greek8
	csISOLatinGreek
Windows-1253 (MIBenum: 2253)	cp1253
	ms1253
	win1253

Hebrew Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-8859-8 (MIBenum: 11)	ISO_8859-8:1988
	ISO_8859-8
	iso8859-8
	8859_8
	ibm916
	ibm-916
	cp916
	916
	iso-ir-138
	hebrew
	csISOLatinHebrew
Windows-1255 (MIBenum: 2255)	win1255
	cp1255
	ms1255

Japanese Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-2022-JP (MIBenum: 39)	iso2022-jp
	iso-2022-jis
	junet
	jis
	jis_encoding
	csJISEncoding
	csISO2022JP
ISO-2022-JP-2 (MIBenum: 40)	iso-2022-jp2
	csISO2022JP2

Preferred MIME Name or Primary Registered Name	Encoding Aliases
Shift_JIS (MIBenum: 17/2024)	sjis shift-jis ShiftJis x-sjis x-shift-jis windows-31j csWindows31J ms932 cp932 win932 windows-932 MS_Kanji csShiftJIS pck \u30b7\u30d5\u30c8\u7b26\u53f7\u5316\u8868\u73fe
EUC-JP (MIBenum: 18)	Extended_UNIX_Code_Packed_Format_for_Japanese eucjp x-euc-jp euc_jpnew 10/18/99 x-eucjp eucjis csEUCPkdFmtJapanese

Korean Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
euc-kr (MIBenum: 38)	euc_kr euckr csEUCKR

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ks_c_5601-1987 (MIBenum: 36)	ks_c_5601-1989
	ksc5601-1987
	ksc5601_1987
	ksc_5601
	ksc5601
	5601
	korean
	csKSC56011987
IBM949 (MIBenum: ????)	ibm-949
	cp949
	cp-949
	949
Windows-949 (MIDenum: ????)	win949
	ms949

Thai Character Set

Preferred MIME Name or Primary Registered Name	Ecoding Aliases
IBM874 (MIBEnum: ????)	ibm-874
	cp874
	874
Windows-874	win874
	ms874

Turkish Character Set

Preferred MIME Name or Primary Registered Name	Encoding Aliases
ISO-8859-9 (MIBenum: 12)	ISO_8859-9:1989
	ISO_8859-9
	iso8859-9
	8859_9
	ibm920
	ibm-920
	cp920
	920
	iso-ir-148
	l5
	latin5
	csISOLatin5
Windows-1254 (MIBenum: 2254)	win1254
	cp1254
	ms1254

Vietnamese Character Set

Preferred MIME Name or Primary Registered Name	Ecoding Aliases
Windows-1258 (MIBenum: 2258)	win1258
	ms1258
	cp1258
	cp-1258