Gateway Customization Guide

Netscape Directory Server

Version 4.1

The Team

Writer: Doug Meier

Technical Editor, Gateway: Rob Powers

Technical Editor, Gateway Localization: Noriko Hosoi

Version 4.1

99 98 97 10 9 8 7 6 5 4 3 2 1

Netscape Communications Corporation 501 East Middlefield Road, Mountain View, CA 94043

Preface

The descriptions, instructions, and examples in this guide can be used to create and modify a gateway instance to suit the needs of users in your organization.

Who Needs This Guide

This guide is for anyone who needs to implement a simple gateway instance with basic directory lookup functionality. It is also for users who wish to implement a more powerful gateway instance with directory authentication and administration capability. This guide assumes the reader is familiar with HTML file syntax and has a rudimentary understanding of how the LDAP directory stores information.

What's In This Guide

- Chapter 1, Gateway Features: Describes features and functionality of the gateways installed with Directory Server 4.x. HTTP server recommendations are also provided.

- Chapter 2, Setting Up the Gateway: Identifies the planning decisions and tasks required to install and initially configure a gateway. Includes instructions for HTTP server configuration, procedures for setting up a new gateway instance, procedures for changing configuration settings on existing gateways, and recommendations for configuring gateway clients.

- Chapter 3, Gateway Localization: Describes Netscape support for UTF-8, explains how the gateway selects a character set, identifies the tasks required to set up additional gateway locales, and explains how to set up gateway locales for translation.

- Chapter 4, Files Controlling Gateway Functionality: Identifies gateway file components, including search configuration files, object class templates, script files, and search result templates.

- Chapter 5, Entry Types and Object Class Attributes: Provides examples of entry types (objects) defined in the gateway's .conf file, provides examples of locations mapped to entry types, and discusses gateway directives.
- Chapter 6, Search Attributes, Filters, and Results: Provides examples for modifying search attributes, adding search support for a new object, modifying default search filters, and customizing search results.
- Chapter 7, Customizing Graphics and Color: Describes how to change the appearance of default gateway forms.
- Appendix A, .conf Parameters: Describes, in alphabetical order, gateway configuration parameters defined in dsgw.conf and pb.conf.
- Appendix B, Gateway Directives Reference: Describes directives used in Gateway HTML object class and search result templates.
- Appendix C, CGI Usage: Provides regular expression-type descriptions of the ways to invoke the CGIs in script files.
- Appendix D, Gateway User Help: Provides instructions for using the Directory Server 4.0 gateway interface. The information supplied here is intended to be repurposed as help topics that can be made available to gateway users from the Help button on gateway forms.

Related Information

For more information about working with Directory Server, see these documents:

- *Netscape Directory Server Installation Guide, Version 4.0*
- *Netscape Directory Server Administration Guide, Version 4.0*
- *Netscape Directory Server Release Notes, Version 4.0*
- *Managing Servers with Netscape Console, Version 4.0*
- *Netscape Directory Server Deployment Guide, Version 3.0*
- *Customizing LDAP Settings for Communicator 4.0x*

Contents

Tables

Examples

Chapter

1

Gateway Features

This chapter describes the gateway features introduced with release 4.0 of the Directory Server. Topics include:

- What Is a Gateway?
- HTTP Server Requirements for Gateways
- Gateways Installed with Directory Server 4.0
- Non-Anonymous Searching
- Compatibility with Existing Gateways
- Automatic Updates to Directory Configuration

What Is a Gateway?

A gateway is an HTTP-to-LDAP client that lives on an HTTP server. Using special directives embedded in HTML files, a gateway allows users to access user directory data using any kind of web browser.

A gateway can be used to perform directory lookup, or to authenticate to the directory and complete a database administration task, without logging in to Netscape Console.

In Version 4.0 of Netscape Directory Server, many gateway instances can be defined on one HTTP server, providing access to any number of Directory Servers. A gateway instance consists of:

- a .conf file, stored in <NSHOME>/dsgw/context, defining the context for a gateway instance (for instance, dsgw.conf defines the *dsgw* gateway instance).
- an HTML directory for object class templates and other files containing gateway directives used to communicate with the LDAP server.
- a configuration directory for directory search, directory authentication, language files, and gateway scripts.

HTTP Server Requirements for Gateways

A gateway instance requires an HTTP server capable of communicating with the LDAP Directory Server.

For optimum performance and highest security, the gateway should be configured to run under a high-performance HTTP server, such as the Netscape Enterprise server.

See Also

"HTTP Server Configuration" on page 17

Gateways Installed with Directory Server 4.0

Two gateway instances are installed during Directory Server installation: the default gateway and Directory Express. The default gateway provides search, authentication, and entry modification capability. Directory Express is a customized version designed specifically for use as a corporate phonebook.

The two gateway instances provided with Directory Server are configured to use the suffix set when the Directory Server was configured, and non-SSL (Secure Socket Layer) communications. Either can serve as a framework for a building a more sophisticated gateway instance.

See Also

"Location of Gateway Files" on page 12

Default Gateway (dsgw.conf)

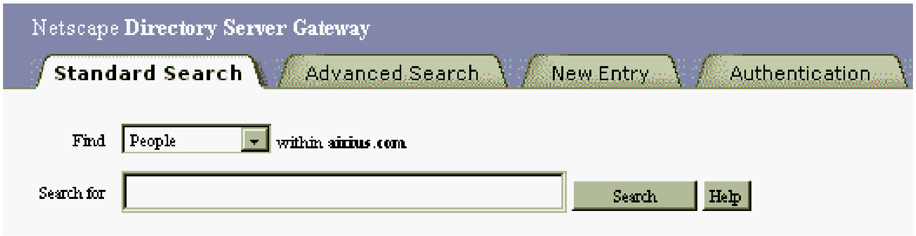
In addition to the standard search form, the default Gateway provides an advanced search form, a Directory Server authentication form, and a form for adding and modifying entries.

The configuration file for the default gateway is at <NSHOME>/dsgw/context/dsgw.conf.

During Netscape Directory Server 4.0 installation, the default gateway is configured to use as its HTTP server the Netscape Administration server installed with the directory. Following Directory Server installation, the default gateway can be accessed from http://adminhost:adminport/ or directly using this URL:

```
http://adminhost:adminport/dsgw/bin/search?context=dsgw
```

Figure 1.1 Default Gateway



Find People within airius.com

Search for

Directory Express (pb.conf)

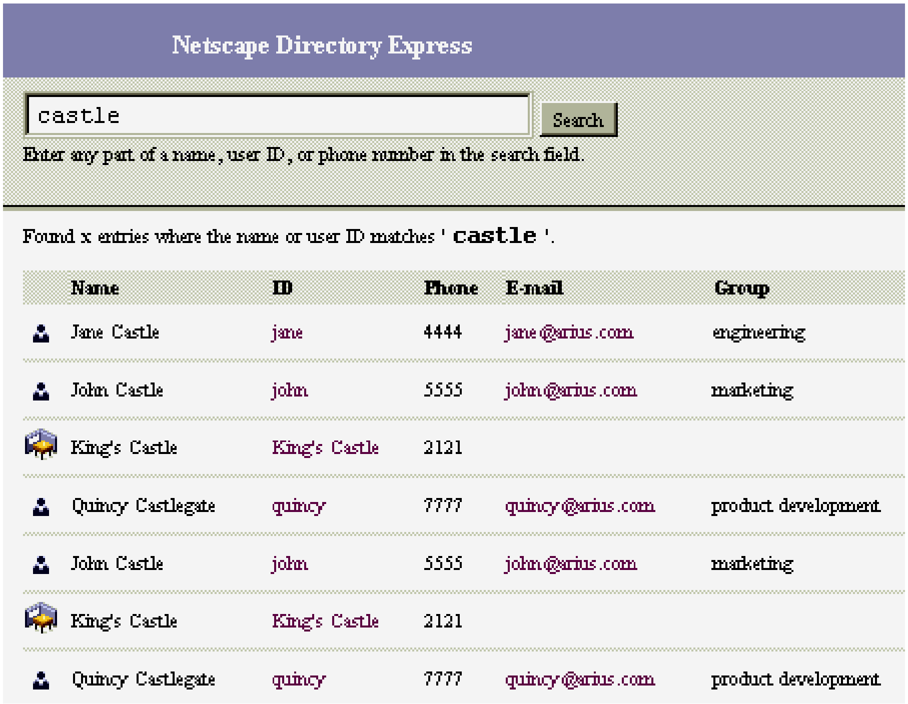
Directory Express is designed for use out of the box as a corporate phonebook. The configuration file for Directory Express is <NSHOME>/dsgw/context/pb.conf.

During Netscape Directory Server 4.0 installation, Directory Express is configured to use as its HTTP server the Netscape Administration server installed with the directory.

Following Directory Server installation, Directory Express can be accessed from http://adminhost:adminport/ or directly using this URL:

```
http://adminhost:adminport/dsgw/bin/lang?context=pb
```

Figure 1.2 Directory Express Search Result



Netscape Directory Express

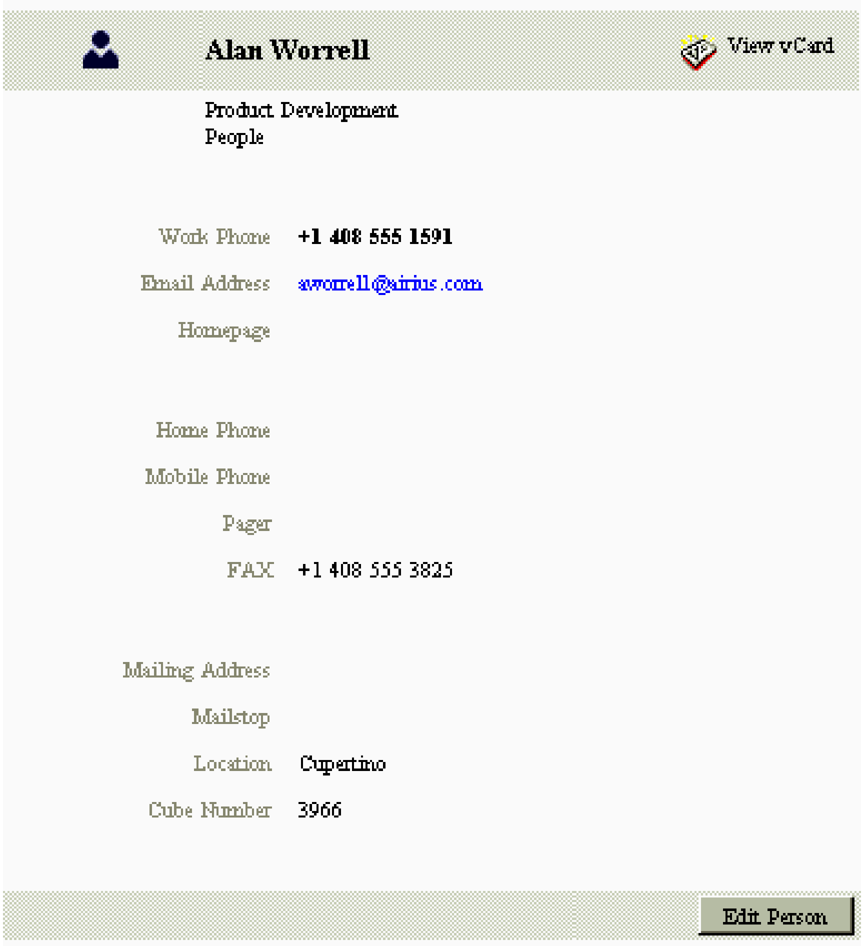
castle

Enter any part of a name, user ID, or phone number in the search field.

Found x entries where the name or user ID matches ' castle '.

Name	ID	Phone	E-mail	Group
Jane Castle	jane	4444	jane@airius.com	engineering
John Castle	john	5555	john@airius.com	marketing
King's Castle	King's Castle	2121		
Quincy Castlegate	quincy	7777	quincy@airius.com	product development
John Castle	john	5555	john@airius.com	marketing
King's Castle	King's Castle	2121		
Quincy Castlegate	quincy	7777	quincy@airius.com	product development

Figure 1.3 Directory Express Extended Search Results



Alan Worrell

Product Development
People

Work Phone	+1 408 555 1591
Email Address	aworrell@airius.com
Homepage	
Home Phone	
Mobile Phone	
Pager	
FAX	+1 408 555 3825
Mailing Address	
Mailstop	
Location	Cupertino
Cube Number	3966

Support for Multiple Gateway Instances

Release 4.0 of the Directory Server supports multiple gateway instances. Many gateways can access directory data from the same HTTP server without conflict.

The .conf files defining the configuration of gateway instances are stored in <NSHOME>/dsgw/context. Within the .conf file are two parameters specifying the path names for the HTML and template files for the gateway (see Example 1.1).

Example 1.1 HTML and configuration directories specified in pb.conf

```
htmldir../pbhtml
configdir ../pbconfig
```

Specifying Gateway Configuration to Gateway CGIs

Information about which .conf file to use is communicated in the QUERY STRING using a GET, and through a hidden variable on a POST.

GET Operations (GCONTEXT)

In a GET operation, gateway CGIs get the gateway context from the QUERY STRING in the URL.

Use the GCONTEXT directive in all URLS to gateway CGIs. Embed <!-- GCONTEXT --> after the CGI name, as shown in Example 1.2. This directive will be replaced by the current CGI's gateway context. The GCONTEXT directive is the only gateway directive that does not have to be at the beginning of the line.

Example 1.2 Embedding GCONTEXT in a Link

```
<a href=/dsgw/bin/lang?<!-- GCONTEXT -->&file=auth.html>click</a>
```

Gateway 3.x Note

In release 3.x, CGIs passed their arguments using PATH_INFO. In 4.0, PATH_INFO is replaced by the QUERY_STRING, which works better with the Japanese version of Windows NT, and is functionally equivalent.

POST Operations (PCONTEXT)

In a POST operation, the CGI posts to the gateway instance specified by a hidden variable on an HTML form. Each POST operation to a gateway CGI in an HTML form must use the PCONTEXT directive so that CGIs can pass the gateway instance to the next page and maintain the state.

For CGI invocations using a POST, put <!-- PCONTEXT --> at the beginning of a line, as shown in Example 1.3.

Example 1.3 Specifying PCONTEXT in an HTML Form

```
<form method=post action=/dsgw/bin/dosearch>
<input type=hidden name=dn value=””>
<!-- PCONTEXT -->
</form>
```

See Also

“Context-Related Directives” on page 93

URLs for Gateway Access

The URL customers use to access a 4.0 gateway differs from the URL customers use to access a 3.x gateway.

Release 4.0

To get to a 4.0 gateway, users enter the following URL:

```
http://adminhost:adminport/dsgw/bin/lang?context=<context>
```

The default gateway and Directory Express are available from the page for the Netscape Administration server (http://host:port)

Release 3.x

To get to a 3.x gateway, users enter the following URL:

```
http://host:port/dshtml
```

Gateway Localization

The gateway is designed to allow support for clients in multiple locales.

Gateway Locales

Release 4.0 of the Directory gateway is localized for English, Japanese, French, and German. Language files are stored in <NSHOME>/dsgw/html/<lang> and <NSHOME>dsgw/config/<lang>, where <lang> is defined in RFC1766 (for example, language files for Japanese are stored in /usr/netscape/dsgw/html/ja and /usr/netscsape/dgw/config/ja[true]). The gateway can be configured to support additional locales.

Support for the character sets necessary to render a particular locale (language) must be available in the browser's configuration. Netscape Communicator provides support for multiple languages and multiple characters sets.

See Also

"Gateway Locales" on page 34
"Configuring Gateway Clients" on page 27

Character Sets

Release 4.0 of the gateway supports Unicode (specifically, the UTF-8 character set), the character set containing all of the world's languages. However, Unicode or any character set specified in the gateway's .conf file can be overridden by the HTTP Accept-charset header sent by the gateway client or by a character set created for a specific language.

See Also

"How the Gateway Selects a Character Set" on page 32

Non-Anonymous Searching

Release 3.0 of the gateway allows anonymous searching only. This provides only the most basic permissions for accessing information in the user directory.

The release 4.0 gateway provides the same functionality but in addition supports non-anonymous searching. A bind DN and bind password, stored in a file named binddnfile, can be set up for users to authenticate to the Directory Server. User permissions for directory access can be defined in the Netscape Console.

If authentication credentials exist for a user on the Directory Server, these override the bind DN and bind password in the gateway's binndnfile. When authentication credentials expire or are invalid, the gateway attempts to authenticate the user to the directory using the binddnfile. When no binddnfile is specified, the gateway instance binds anonymously.

binddnfile

The location of the binddnfile containing bind DNs and bind passwords for individual users and groups of users is specified in the gateway's .conf file.

See Also

"binddnfile" on page 81

Do Not Server Up binddnfile over HTTP

The binddnfile contains highly sensitive information. Do not store it under <NSHOME>/dsgw or in any directory that is served up over HTTP (for instance, /bin/slapd/server is a good place to store the binddnfile).

Compatibility with Existing Gateways

Release 4.0 of the Directory Server is compatible with previous versions of the gateway, including release 3.x. Although the HTML files and gateway CGIs are not interchangeable, a 3.x gateway can access a 4.0 Directory Server, and a 4.0 gateway can access a 3.x Directory Server.

It is not necessary to upgrade a 3.x gateway to release 4.0, however a script is available to complete this task. On UNIX and NT systems, this migration occurs automatically when a Directory Server is migrated from release 3.x to release 4.0.

The gateway migration script is described in "Migrating 3.x default Gateways to Release 4.0" on page 14.

Directory server migration is described in the *Release Notes* for Directory Server 4.0

Automatic Updates to Directory Configuration

A script that updates gateway instances with changes to Directory Server configuration, *updatedsgw*, is included with the Directory Server installation. This script searches <NSHOME>/dsgw/context for gateway instances that match the Directory Server host and port.

The updatedsgw script runs automatically for gateways installed on the Netscape Administration server managing the Directory Server instance. When the server port or root DN (or other settings, such as directory manager) for a Directory Server instance is changed, the Netscape Administration server managing the Directory Server instance runs the updatedsgw script.

See Also

"Updating the Gateway with Changes to Directory Server Configuration" on page 15

Chapter

2

Setting Up the Gateway

This chapter describes the planning decisions and tasks required to install and initially configure a gateway for access by end users. Topics include:

- Gateway Installation Planning
- HTTP Server Configuration
- Creating a New Gateway Instance (4.0 only)
- Gateway .conf File Configuration
- Configuring Gateway Clients

Gateway Installation Planning

When preparing to deploy a gateway in the enterprise, consider the following planning issues:

- where to store gateway configuration and HTML files
- how to securely store gateway bind DNs and bind passwords
- how to protect root processes on the HTTP server running the gateway

- whether to migrate existing gateways to release 4.1 gateways or to let them co-exist
- how changes to Directory Server configuration and user directory will be updated on the gateway
- which type of HTTP server best suits the needs of your enterprise

Location of Gateway Files

Table 2.1 and Table 2.2 show the locations of gateway files for release 4.0 and release 3.x.

Gateway Release 4.0

Two gateway instances are installed during Directory Server 4.0 installation: Directory Express and the default gateway. The .conf files (pb.conf and dsgw.conf) are stored in /dsgw/context. Additional gateways can be created by customizing Directory Express or the default gateway.

Table 2.1 Location of gateway files for release 4.0

File Type	Location
default gateway .conf file	<NSHOME>/dsgw/context/dsgw.conf
default gateway (dsgw) HTML and template files	<NSHOME>/dsgw/html <NSHOME>/dsgw/config
Directory Express .conf file	<NSHOME>/dsgw/context/pb.conf
Directory Express (pb) HTML and template files	<NSHOME>/dsgw/pbhtml <NSHOME>/dsgw/pbconfig

Gateway Cloning

Unique gateway instances may have unique HTML directories (for example, ..dsgw/mythml) and template directories (/for example, ..dsgw/myconfig). However, gateways may also be cloned to use identical HTML and template directories while pointing to different Directory Servers or different suffixes on a Directory Server.

See Also

"Gateway Cloning" on page 23

Gateway Release 3.x

One gateway is installed for each instantiation of the Directory Server. The .conf file is stored in /dsgw (under slapd-<instance>).

Table 2.2 Location of gateway files for release 3.x

File Type	Location
default gateway .conf file	<NSHOME>/slapd-<server>/dsgw/dsgw.conf
default gateway HTML and template files	<NSHOME>/slapd-<server>/html <NSHOME>/slapd-<server>/config

Securing 4.0 Gateway Configuration and Settings

Protecting Bind DN and Password

Release 4.0 gateway .conf files reference files that contain sensitive information, including the binddnfile containing the bind DN and bind password used to permit non-anonymous searching of the directory. The binddnfile should not be stored under the gateway configuration directory (<NSHOME>/dsgw), or in any directory that is served up over HTTP.

Protecting Root Processes on UNIX Systems

On UNIX systemsm, it is not advisable to run the gateway from a Netscape Administration server that is also running a Netscape server process as root. This may expose sensitive information about the configuration of Netscape servers.

Migrating 3.x default Gateways to Release 4.0

Release 4.0 of the Netscape Directory server supports 3.x gateways, so it is not necessary to modify a 3.x gateway in order for it to access a 4.0 directory. However, a 3.x gateway can not co-exist with 4.0 gateways unless it is first migrated to the 4.0 gateway structure.

An upgrade script, *dsgwmig*, is available which completes the migration of dsgw.conf and its relevant files. The script does not overwrite the existing 3.x gateway, but renames it as a new gateway instance (dsgw-30.conf) that runs from the 4.0 /dsgw installation directory. The 3.x gateway will continue to work with the 3.x Directory Server installation.

The migration script is installed in <NSHOME>/bin/slapd/admin/bin during Directory Server installation. The script makes the following changes to the 3.x gateway.

- Replaces /ds with /dsgw/bin in the gateway's URLs.
- Changes gateway HTML file syntax to support the GCONTEXT and PCONTEXT directives.
- Adds parameters to the dsgw .conf file--including htmldir, configdir, gwnametrans--that support multiple gateway instances.
- Adds localization parameters to dsgw.conf to support the UTF-8 character set and to support older browsers that are not aware of UTF-8.
- Moves dsgw.conf to /<NSHOME>/dsgw/context and renames it dsgw-30.conf.
- Changes PATH_INFO to QUERY_STRING

Location of migrated dsgw.conf file

When the migration script finishes, the migrated 3.x dsgw.conf file can be accessed from:

```
http://<host>:<port>/dsgw/bin/lang?context=dsgw-30
```

Running the Gateway Migration Script

The dsgwmig script runs automatically during the 3.x -> 4.0 Directory Server migration process.

Directory server migration is described in the *Release Notes* for Directory Server 4.0.

Updating the Gateway with Changes to Directory Server Configuration

Release 4.0 of the Directory Server provides a script, *updatedsgw*, that can be used to update all gateway instances with changes to the Directory Server configuration, including changes to Directory Server port, host, suffix, and root DN (the ability to update the suffix is not available in the server administration console). The updatedsgw script is stored in <NSHOME>/bin/slapd/admin/bin.

Changes made to the Directory Server configuration (slapd.conf) through the Netscape Console are posted to updatedsgw and the relevant gateway files are updated. These files are updated when the host and port for the gateway match the host and port of the Directory Server.

Note The Directory Server's root DN (the Directory Server's superuser) must match the value of the gateway's dirmgr parameter.

HTTP Server Recommendations for Directory Server Gateway Release 4.0

The Netscape Administration Server is the default HTTP server for the two gateway clients that are installed with the Netscape Directory Server. Both Directory Express and the default gateway are preconfigured to run under the Netscape Administration Server without additional setup.

Factors Affecting Gateway Usage

Many factors affect gateway performance on an HTTP server, including:

- the number of users accessing the gateway at a given time
- the complexity of the directory searches performed and the search results required
- whether the gateway is additionally to be used for authentication and login
- the load from other processes managed by the host machine
- the speed and performance of the computer hardware selected for the host computer
- the speed and capacity of the network (network hardware and software)

In general, gateway performance on the Netscape Administration Server begins to slow down when the number of users accessing the gateway throughout the enterprise reaches 6,000 people. (Note that this is a very general recommendation that does not take into account factors listed above, especially the speed of the host machine.)

Note It is not advisable to run the gateway from a Netscape Administration server that is also running a Netscape server process as root. This may expose sensitive information about the configuration of Netscape servers.

Running the Gateway in High-Usage Networks

Network administrators expecting high gateway usage may want to move the gateway to a high-performance HTTP server that is dedicated to running the gateway.

Note If you do decide to migrate the gateway's configuration files to a high-performance HTTP server, Netscape recommends Netscape Enterprise Server or FastTrack Server.

See Also

"Name Translation Mapping" on page 17

HTTP Server Configuration

Name Translation Mapping

The HTTP server uses Name Translation mapping to translate a virtual path provided by a gateway client to a physical path used by an HTTP server. This Name Translation mapping specifies the gateway's HTML directory. The gateway's CGIs use this information to output the correct URL (HTTP redirection). In release 4.0 of the gateway, the NameTrans mapping is specified in the gateway's .conf file using the gwnametrans parameter.In release 3.x, the NameTrans mappings are hard-coded (the binary files stored in /dsgw/bin are mapped to /dshtml and the HTML files stored in /dsgw/html are mapped to / ds).

See Also

"gwnametrans" on page 83

Gateway Root Suffix

In release 4.0, Directory Express and the default gateway are set to the root suffix specified during Directory Server installation. This suffix specifies the DN for the LDAP database and represents a root in the directory tree (for example, o=airius.com). Multiple gateways can be set up on an HTTP server that provide access to directory entries that correspond to this root suffix.

When the Directory Server's suffix changes, it is necessary to run the updatedsgw script manually in order to propagate the change to all gateway instances.

See Also

"Updating the Gateway with Changes to Directory Server Configuration" on page 15

In release 3.x, to access a different suffix, another HTTP server and another gateway must be configured on the host where the Directory Server is installed.

Note When the root suffix, directory manager, or port change, the gateway settings in dsgw.conf must be updated to reflect the changes (if they haven't been updated through the Netscape Console).

Configuring the Gateway for Enterprise or FastTrack Servers

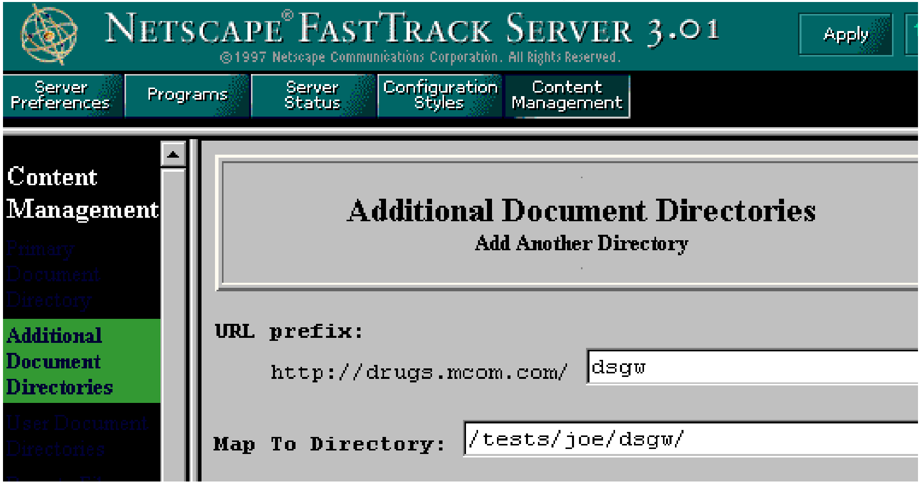
The configuration procedures provided in this section assume that a Netscape FastTrack (or Enterprise) HTTP server is installed and configured to communicate with the Netscape LDAP Directory server.

In release 4.0, Directory Express and the default gateway are installed with the Directory Server and configured to run under the Netscape Administration Server, which is the default HTTP server for the gateway clients. No additional configuration is necessary. However, customers in high-usage networks may wish to move their gateways (or set up new gateways) on a high-performance HTTP server, following the procedures provided in this section.

In release 3.x, the gateway must be configured to communicate with the LDAP Directory Server, following the procedures provided in this section. The 3.x gateway allows one instance of the gateway per HTTP server.

Figure 2.1 shows the Netscape FastTrack Server configuration screen used to configure an additional document directory. Figure 2.2 shows the Netscape Enterprise Manager configuration screen used to configure an additional CGI directory.

Figure 2.1 Configuring an Additional Document Directory

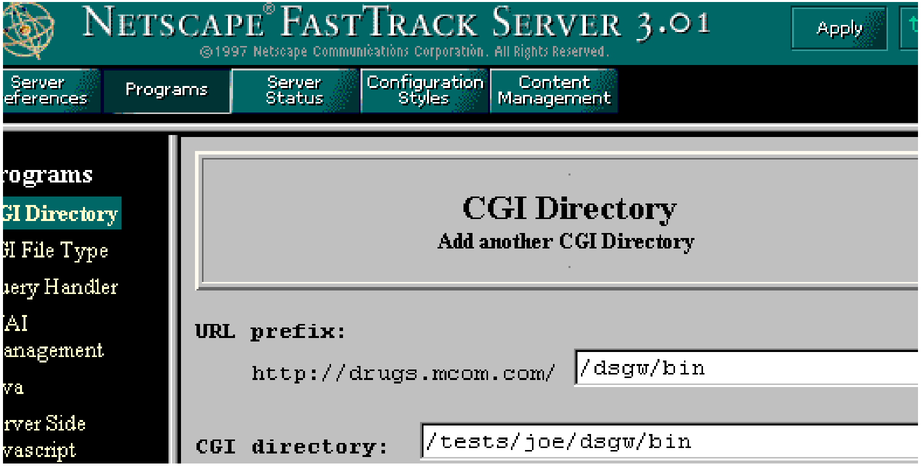

NETSCAPE® FASTTRACK SERVER 3.01
©1997 Netscape Communications Corporation. All Rights Reserved.
Apply
Server Preferences
Programs
Server Status
Configuration Styles
Content Management
Content Management
Primary Document Directory
Additional Document Directories
User Document Directories

Additional Document Directories
Add Another Directory

URL prefix:
http://drugs.mcom.com/ dsgw

Map To Directory: /tests/joe/dsgw/

Figure 2.2 Configuring an Additional CGI Directory



CGI Directory
Add another CGI Directory

URL prefix:
http://drugs.mcom.com/ /dsgw/bin

CGI directory: /tests/joe/dsgw/bin

Add an Additional Document Directory (4.0 Gateways)

Adding an additional document directory is necessary to establish access to the gateway files. From the server manager for the FastTrack or Enterprise Server:

1. Go to Content Mgmt | Additional Document Directories.

2. In the URL prefix field, enter

   ```
   dsgw
   ```

3. In the Map to Directory field, enter

   ```
   <NSHOME>/dsgw/
   ```

   where <NSHOME> is the Directory Server's installation directory.

4. Click OK, then Save and Apply.

Add an Additional Document Directory (3.x Gateways)

Adding an additional document directory is necessary to establish access to the gateway files.

From the server manager for the Enterprise Server:

1. Go to Content Mgmt | Additional Document Directories.

2. In the URL prefix field, enter dshtml.

3. In the Map to Directory field, enter:

   ```
   <NSHOME>/slapd-<serverID>/dsgw/html
   ```

   where <NSHOME> is the Directory Server's installation directory, and <serverID> is the Directory Server's identifier.

4. Click OK, then Save and Apply.

Add an Additional CGI Directory (for 4.0 gateways)

Adding an additional CGI directory is necessary to make the gateway's CGI programs available. From the server manager for the HTTP server:

1. Go to Programs | CGI Directory.
2. In the URL prefix field, enter

   ```
   /dsgw/bin
   ```
3. In the CGI directory field, enter:

   ```
   <NSHOME>/dsgw/bin
   ```
4. Click OK, then Save and Apply.

Add an Additional CGI Directory (for 3.x Gateways)

Adding an additional CGI directory is necessary to make the gateway's CGI programs available. From the server manager for the HTTP server:

1. Go to Programs | CGI Directory.
2. In the URL prefix field, enter ds.
3. In the CGI directory field, enter

   ```
   <NSHOME>/slapd-<serverID>/dsgw/bin
   ```
4. Click OK, then Save and Apply.

Change Permissions of Cookie Directory (UNIX only)

To enable the gateway to store cookies on the HTTP server, the gateway must have write access to the HTTP server's cookie directory. From the server manager for the HTTP server:

1. Go to System Settings|View Server Settings and note the value set for the User field.

   If this value is set to nobody, check to make sure that the server is not running as a named user. For example, on Solaris grep for the http process:

   ```
   ps -ef | grep http
   ```

The process listed identifies the name under which the HTTP process is running.

2. Log into the machine as root.

3. Go to the <NSHOME>/dsgw and enter:

```
# chown <uid> authck
```

where <uid> is the user name determined in step 1.

4. Verify that the directory is accessible by opening the URL:

```
http://<webserver-host>:<port>/ds/search
```

where <webserver-host> is the HTTP server's host name, and <port> is the port number used by the server.

Note When the HTTP server is using the standard HTTP port number (80), the port number does not need to be included in the URL.

Creating a New Gateway Instance (4.0 only)

These instructions assume that the new gateway instance is running under the Netscape Administration server or a similarly capable HTTP server.

Example 2.1 Setting up .conf file and directories for new gateway instance

1. Rename dsgw.conf or pb.conf to a new gateway context. For example, dsgw/context/dsgw.conf might become dsgw/context/airius.conf.

2. Set the gwnametrans parameter in the new gateway's .conf file to point to the HTML directory. For example, the gwnametrans parameter setting for airius.conf should point to /dsgw/airiushtml.

3. To support non-anonymous searching (one individual user DN and password per directory instance) using the new gateway, set the binddnfile parameter in airius.conf to point to the location of the file containing the bind DN and bind password be used to access information in the user directory (the binddnfile contains sensitive information; for security purposes, do not store the binddnfile within the /dsgw directory or within any directory served up over HTTP).

4. Create an HTML directory for the new gateway: For example, to provide an HTML directory for airius.conf, copy and rename an existing HTML directory (dsgw/html or dsgw/pbhtml) to /dsgw/airiushtml.

5. Create a template directory containing object class templates and other configuration files. For example, to provide a template directory for airius.conf, copy and rename an existing template directory (/dsgw/config or /dsgw/pbconfig) to /dsgw/airiusconfig.

6. Edit the htmldir and configdir parameters in airius.conf to point to the new HTML and template directories.

7. To access the new gateway instance--in this example, airius.conf--navigate the browser to

   ```
   http://admin:port/dsgw/bin/lang?context=airius
   ```

Gateway Cloning

The HTML and template directories for one gateway can serve as the HTML and template directory for many others. Maintaining the functionality of multiple gateways in a centralized /config and /html directories is useful when the only values that are likely to change are parameter setting in the .conf file. (for example, the host and port specified by the baseurl parameter, the root DN specified by dirmgr, and the root suffix specified by the location-suffix parameter).

Gateway .conf File Configuration

Changing the Default Port Setting

The LDAP port is set during Directory server installation. This value can be changed in the baseurl parameter. Example 2.2 shows the syntax used to specify a port number that is different than the default port number of 389.

Example 2.2 Changing LDAP port in the baseurl parameter

```
baseurl “ldaps://dirserver.airius.com:3000/o%3Dairius.com”
```

Setting Up the Directory Manager

When the Directory server is installed, the Directory Manager is by default set to the root DN. The Directory Server 4.0 installation requires a root DN. If no root DN was configured when the Directory server was installed, then no default Directory Manager is configured for the gateway.

Note For security reasons, Netscape recommends setting the Directory Manager to an entry other than the root DN.

Configuring the Directory Manager DN

Use this procedure to configure the gateway Directory Manager to reference the correct DN.

1. Create an entry for the Directory Manager, making sure to set a password for the entry.
2. Set the permissions for the Directory Manager so that it has read and write authority for the entries it manages.
3. When necessary, change the dirmgr parameter to refer to the Directory Manager's distinguished name.

Note End users frequently forget their passwords, so give the Directory Manager write access to the userPassword attribute for the entries it manages.

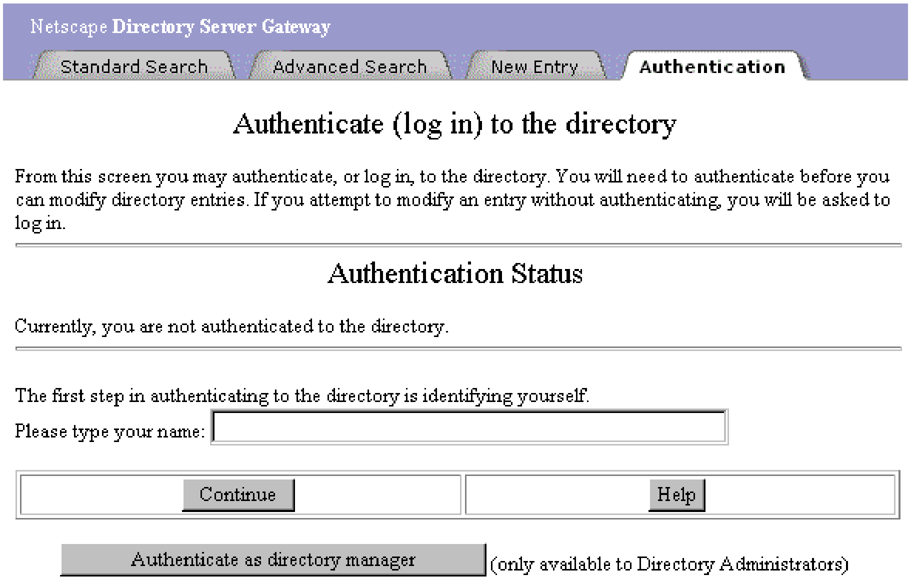
The dirmgr parameter is described in "dirmgr" on page 82. Creating directory entries is described in the *Netscape Directory Server Administrator's Guide*.

Authenticating as Directory Manager

Figure 2.3shows the authentication login screen for the default gateway. Administrators can use it to authenticate as the Directory Manager. The Authenticate as Directory Manager button is displayed only when a Directory Manager has been configured for the gateway.

The authlifetime parameter, which defines the number of seconds that a user may remain authenticated, is described in "location" on page 84.

Figure 2.3 Authenticating as Directory Manager


Netscape Directory Server Gateway
Standard Search
Advanced Search
New Entry
Authentication
Continue
Help
Authenticate as directory manager

Authenticate (log in) to the directory

From this screen you may authenticate, or log in, to the directory. You will need to authenticate before you can modify directory entries. If you attempt to modify an entry without authenticating, you will be asked to log in.

Authentication Status

Currently, you are not authenticated to the directory.

The first step in authenticating to the directory is identifying yourself.

Please type your name:

(only available to Directory Administrators)

Setting up the Suffix for Adding Entries

The location-suffix parameter is defined in dsgw.conf, and identifies the suffix under which the gateway creates new entries in the directory. The location-suffix parameter can point to any suffix in a directory.

Setting the location-suffix parameter is described in “include” on page 84. The *Netscape Directory Server Administrator’s Guide* describes the Suffix parameter and provides syntax examples. Setting the root suffix is also described in the *Netscape Directory Server Installation Guide*.

Setting Up SSL Support

When the Directory server is installed, the gateway is configured to communicate with the Directory server using a non-SSL host name and port number. This information is stored in the baseurl parameter.

Configuring the gateway to use SSL when communicating with the Directory server requires modification of the securitypath and baseurl parameters in dsgw.conf.

Enabling SSL communications on the Directory Server is described in the *Netscape Directory Server Administrator's Guide*. Information about managing key and certificate databases is provided in *Managing Netscape Servers*.

Configuring the Gateway to Use SSL

The syntax in Example 2.3 shows the securitypath parameter specifying the location of the certificate database.

Example 2.3 Specifying the Path for Certificate Database

```
securitypath "/export/TEST/alias/cert.db"
```

The syntax in Example 2.4 shows the baseurl parameter configured to use ldaps (instead of ldap, the default) and standard SSL port number 636.

Example 2.4 Specifying SSL Communication

```
baseurl "ldaps://dir.airius.com:636/o%3Dairius.com"
```

Note Before configuring SSL, verify that the gateway's Certificate database contains a server certificate or Certificate Authority certificate needed to communicate with the Directory Server.

See Also

"baseurl" on page 80

Setting Up Localization

There are two considerations for configuring the gateway character set: the directory contents and the HTTP clients. The ideal character set supports all the characters in the directory, and is displayed properly by all HTTP clients. UTF-8 best supports the Directory Server's internal character (which is UTF-8). However, HTTP clients that are not designed for localization may display UTF-8 poorly.

If a single character set works well for most gateway users, define it using the charset parameter in the gateway's .conf file. For users who require a character set that supports another language, create the appropriate ../dsgw/LANG/dsgwcharset.conf file (where LANG represents a language, such as "en" or "fr") and configure the HTTP clients for these users to specify their language in the HTTP Accept-language header.

Setting the language and character set for communication with HTTP clients is described in Chapter 3, "Gateway Localization"

Setting Locations and newtypes

See Also

"Mapping Locations and Entry Types" on page 48

Setting vCard Properties

Mappings between VCARD properties and LDAP attribute type are described in "vcard-property" on page 87.

Configuring Gateway Clients

Language Support for HTTP Clients

When a user accesses information in the directory from an HTTP client—through the gateway or another HTTP-based LDAP interface—the client provides the Directory server with information indicating the optimal character set and collation order to use in transmitting information to the browser.

Unicode and Latin-1 Character Sets

When the user is using Netscape Communicator 4.x, the Directory server sends Unicode characters. Netscape Navigator 2.x and 3.x clients are not capable of displaying Unicode characters.

When the user is using Netscape Navigator 3.x and lower, the Directory server sends Latin-1 characters unless the charset parameter has been configured in the gateway's .conf file. The Latin-1 character set includes most Western languages, including German, French, English, and Spanish.

Displaying a Non-English Alphabet

To display directory content that uses a non-English alphabet, a font capable of displaying a non-English alphabet must be installed on the user's system.

The Directory server can store any Unicode character, so Communicator users should install a font that supports all of Unicode. Bitstream Cyberbit, which is bundled with Netscape Communicator, supports Unicode.

Users who are not using Communicator should use a font that supports Latin-1 (or Western) character sets. Most of the commonly used fonts (Courier, Times Roman, Helvetica) have a Latin-1 variant.

Configuring Communicator 4.x for Preferred Language

1. Install a font that supports Unicode.
2. Go to Edit | Preferences | Appearance | Fonts.
3. From the For the Encoding pull-down menu, select Unicode.
4. From the Variable Width Font pull-down menu, select a Unicode font set (for example, Bitstream Cyberbit).
5. From the Fixed Width Font pull-down menu, select a Unicode font set (for example, Bitstream Cyberbit).

6. Go to Edit | Preferences | Navigator | Languages and configure the list of languages so that the best description of the user's language is first, followed by other acceptable languages. For example, a speaker of British English who also reads Spanish might list English/United Kingdom [en-GB] first, followed by English [en] and then Spanish [es].

Configuring Navigator 3.01 for Preferred Language

1. Install a font that supports Unicode.
2. Go to Options | General Preferences | Fonts.
3. From the For the Encoding pull-down menu, select (on NT) Latin-1 or (on UNIX) Western (ISO-8559-1).
4. For Use the Proportional Font, select a Latin-1 font set.
5. From the Fixed Font pull-down menu, select a Latin-1 font set.
6. Go to Options | General Preferences| Languages and configure the list of languages so that the best description of the user's language is first, followed by less-exact descriptions and other acceptable languages. For example, a speaker of British English who also reads Spanish might list English/United Kingdom [en-GB] first, followed by English [en] and then Spanish [es].

Customizing Communicator's LDAP Settings

Administrators can reconfigure Javascript preference settings in Communicator to allow users to interact with information stored in the user directory.

- In the Address Book and Select Address dialog boxes (accessible from the mail composition window), users can enter one string of search criteria to search an LDAP directory for matching names.

- In the Search Directory dialog, users can enter more complex query expressions to search an LDAP directory using native LDAP searches.
- Users can enter LDAP URLs (beginning with the “ldap://” prefix) in Navigator (web browser) windows to search an LDAP directory.

See Also

Customizing LDAP Settings for Communicator 4.0x

Chapter

3

Gateway Localization

This chapter describes gateway localization and identifies the tasks required to set up additional gateway locales. Topics include:

- Unicode and Netscape Support for UTF-8
- How the Gateway Selects a Character Set
- Special Characters
- Gateway Locales
- Setting Up Locales for Translation

Unicode and Netscape Support for UTF-8

Unicode is a character set containing all the characters of all the world's major languages. There are different standard encodings of Unicode, including UCS-2, which is NT's Unicode version, and UTF-8, the version of Unicode specified by version 3 of the LDAP protocol.

Netscape products use UTF-8 in versions 2 and 3 of LDAP. Most software included in the Directory Server product uses UTF-8 internally, and at interfaces other than LDAP (for example in command-line parameters and LDIF files).

The NT Synchronization Server, installed with the Directory Server, converts UTF-8 to and from NT's Unicode representation (UCS-2).

Note Netscape Communicator 4.x supports UTF-8. Netscape Navigator 3.x does not.

How the Gateway Selects a Character Set

The gateway can output web pages in many character sets. The gateway selects a character set for each HTTP client based on a combination of input from the client and from the gateway's configuration files. Releases 3.x and 4.0 of the gateway select a character set for transmission according to this priority:

- character set defined in the client's HTTP Accept-charset header (in release 4.0, this can be overridden for a particular browser using the ignoreAcceptCharsetFrom parameter).
- character set defined in the client's HTTP Accept-language header (for instance, for Japanese, the character set would be defined as ../dsgw/ja/dsgwcharset.conf)
- character set defined in the gateway's .conf file by the charset parameter.

How the Gateway Selects from Multiple Requested Character Sets

When a client includes more than one character set in a request header, and the gateway supports more than one of these, it selects a character set according to this priority:

- UTF-8
- of the possible character sets, the character set with the highest Q value (for example, "de;q=1, en;q=0.5, fr;q=0.7" would give German the highest Q value)
- the character set that appears first in the request header.
- Latin-1 (ISO-8859-1)

HTTP Clients that Request UTF-8

Browsers designed for localization are configured to request the UTF-8 character set by default. To support localization, the gateway is preconfigured to transmit the UTF-8 character set to these clients: Netscape Communicator version 4.0 and greater and to Internet Explorer version 4.0 and greater. Release 4.0 of the gateway allows this preconfiguration to be overridden using the ignoreAcceptCharsetFrom parameter.

See Also

"ignoreAccetpCharsetFrom" on page 83

The conversion from UTF-8 to the Gateway client's chosen character set is performed shortly before output.

HTTP Clients that Do Not Request UTF-8

For browsers that do not request UTF-8 by default (including Netscape Navigator 3.x and pre-4.0 releases of Internet Explorer), the Gateway selects a character set from the Accept-Charset request header or from the Accept-Language request header, depending on the HTTP client.

Some HTTP clients don't request any character set information. For these clients, the Gateway's charset parameter definition is the default. When the charset parameter is not defined in dsgw.conf, the Gateway uses Latin-1 (which is the default in HTTP).

In addition to UTF-8 and Latin-1, the Gateway can convert to and from several national character sets, depending on the client's needs and configuration, including:

- Shift_JIS
- Big5
- EUC-KR

Special Characters

Non-breaking Space

If the client's character set lacks a character for non-breaking space, but has ideographic space, non-breaking spaces are converted to ideographic spaces before character set conversion.

See the changeHTML directive, in the Gateway configuration file dsgw.conf.

Query Strings

When the Gateway needs to embed a UTF-8 string in an URL, it encodes it in a query string (the query string is the part of the URL that follows the question mark).

This works around a problem with Japanese NT, which garbles environment variables that are in UTF-8 (or any character set except Shift_JIS). The web server passes information to the Gateway CGI programs in environment variables, but the query string environment variable $QUERY_STRING is URL-encoded, so it can handle UTF-8 (from NT's point of view, it's ASCII).

Gateway Locales

The Gateway's default language is US English. Release 4.0 of the Directory Server Gateway interface is also translated into the following locales:

- Japanese
- German
- French

Support for Multiple Locales

A single Gateway instance supports clients in multiple locales concurrently.

Support for multiple locales is accomplished by translating documentation (including online help), the string resource database, and the configuration and HTML template files. A single copy of the compiled code handles all supported locales.

Locale-dependent information is stored in translated files stored in subdirectories identifying the locale name. These editable files are stored separately from the Gateway code. For example, the German translation of config/search.html is stored in config/de/search.html, the French translation is stored in config/fr/search.html, and the Japanese translation is stored in config/ja/search.html.

Setting Up Locales for Translation

The default gateway can be configured to support locales in addition to English (the default locale), French, German, and Japanese. This is part of the overall localization effort, which includes localizing all the configuration and HTML files, including the online help and the string resource database.This is made possible by including a pointer to the mapping table in dsgw-l10n.conf, which is stored during Directory server installation in NS-HOME/dsgw/config/<lang>.

dsgw-l10n.conf

dsgw-l10n.conf provides translation in the Search and Advanced Search pull-down menus for the default Gateway (dsgw.conf). If dsgw-110n.conf is not present in the /config/<lang> directory, translation of the UI does not occur and English characters appear in the pull-down menus for Standard Search and Advanced Search.

Example 3.1 shows how to create a new locale using Chinese as the language for translation.

Example 3.1 Creating a locale for Chinese (zh) translation

1. Create a “zh” directory in NS-HOME/dsgw/context
2. Copy dsgw.conf to the NS-HOME/dsgw/context/zh
3. uncomment this line from the Gateway’s .conf file:

```
include “../config/dsgw-l10n.conf”
```

4. create a “zh” directory in NS-HOME/dsgw/config

5. Copy or create the file dsgw-l10n.conf, stored during Gateway installation in NS-HOME/dsgw/config/<lang>, to NS_HOME/dsgw/config/zh

Note If you are using the US version of the Gateway, dsgw.conf contains a sample of dsgw-l10n.conf.

Chapter 4

Files Controlling Gateway Functionality

This chapter describes gateway components. Topics covered include:

- An Overview of Gateway File Types
- Gateway .conf Files
- Gateway Search Configuration Files
- Object Class Templates
- Gateway Script Files
- Gateway Search Result Templates
- Banner Files

An Overview of Gateway File Types

To the gateway user, the gateway is a set of HTML forms that can be accessed from a web browser to communicate directly with the LDAP Directory Server over HTTP.

To the gateway administrator, the gateway is controlled by a set of files installed during Directory Server installation. These files can be edited to

- create new gateway instances
- edit the object class attributes that define the entries users can add to the Directory
- edit the search object class attributes that define how people search for and view entries in the LDAP directory
- change the appearance of gateway forms
- update the gateway with changes to Directory Server configuration

Files that control gateway functionality are described in Table 4.1.

Table 4.1 Gateway File Types and Locations

File Types	What They Do	Location
Gateway .conf files	Define basic configuration parameters for the gateway and specifies the HTML and template directory.	<NSHOME>/dsgw/context
Gateway search configuration files	Define how the gateway performs searches in the Directory.	<NSHOME>/dsgw/config
Gateway search result templates	Define the filters used to display search results to the user.	<NSHOME>/dsgw/config
Object class templates	Define object classes and attributes for entry types.	<NSHOME>/dsgw/config
Gateway script files	Contain scripts used to communicate instructions to the Directory Server over HTTP.	<NSHOME>/dsgw/config
Banner files	Define appearance of colors, background, graphics on gateway forms.	<NSHOME>/dsgw/html

Gateway .conf Files

A gateway's .conf file describes the gateway's instance. A gateway's .conf file controls the host, port, root suffix, and Directory Manager to use when communicating with the Directory Server. It also controls

- the localization settings for the gateway
- the locations where new entries can be created within the directory
- the types of entries that can be created
- the search base
- whether the gateway uses SSL communications

dsgw.conf

The configuration file for the default gateway, dsgw.conf, is invoked at

```
http://<adminhost:adminport>/dsgw/bin/lang?context=dsgw
```

pb.conf

The configuration file for Directory Express, pb.conf, is a gateway customized as a directory lookup tool. pb.conf is invoked at

```
http://<adminhost:adminport>/dsgw/bin/search?context=pb
```

See Also

Gateway configuration parameters are defined in Appendix A, ".conf Parameters."

Gateway Search Configuration Files

Gateway search configuration files determine how the gateway queries information in the Directory and returns search results to users. gateway search configuration files are stored in the dsgw/config directory. All gateway instances for a Directory Server use these common search configuration files.

dsgwsearchprefs.conf

The object classes and object class attributes than can be searched are specified in dsgwsearchprefs.conf.

See Also

dsgwsearchprefs.conf is discussed in Chapter 6, “Search Attributes, Filters, and Results.”

dsgwfilter.conf

The search filters used to search for a particular object class are specified in dsgwfilter.conf. The gateway uses this file when performing a standard search operation.

See Also

dsgwfilter.conf is discussed in Chapter 6, “Search Attributes, Filters, and Results.”

Object Class Templates

The gateway contains a template file for each object class defined in the gateway. To modify how the gateway displays an entry type, edit the corresponding template file. To add gateway support for a new object class, create a new template file, or modify an existing one.

See Also

Modifying template files is discussed in Chapter 5, “Entry Types and Object Class Attributes.”

Default gateway object class templates stored in the dsgw/conf directory are listed in Table 4.2.

Table 4.2 Default Template Files and Related Object Classes

Template File	Related Object Class
display-country.html	country
display-group.html	groupOfNames
display-groupun.html	groupOfUniqueNames
display-ntgroup.html	ntGroup
display-ntperson.html	ntUser
display-org.html	organization
display-orgperperson.html	inetOrgPerson
display-orgunit.html	organizationalUnit
display-person.html	person

Gateway Script Files

Script files are used to dynamically generate HTML forms for the user, based on information supplied by the user and information retrieved from the Directory Server. Script files contain directives the gateway uses to construct the HTML for a page.

Script files can be modified to change the appearance of text that appears in fields, buttons, and prompts on gateway forms.

Script files used to modify the information provided on gateway forms are identified in Table 4.3.

Table 4.3 Gateway Forms and Corresponding Script Files

Set of forms	Corresponding script files
authentication forms	authPassword.html authSearch.html
standard search forms	searchString.html
advanced search forms	csearchAttr.html csearchBase.html csearchMatch.html csearchString.html csearchType.html
newentry forms	newentryType.html newentryName.html

Gateway Search Result Templates

Search result templates control how the results of a standard or advanced search are displayed when more than one entry is found. A separate search result file exists for each search object defined in dsgwsearchprefs.conf.

The default search result files installed in the dsgw/config directory during Directory Server installation are shown in Table 4.4.

Table 4.4 Default Search Result Templates

Search Result Template	Related Object Class
list-Anything.html	Anything
list-Auth.html	Auth (internal)
list-Groups.html	Groups
list-NT-Groups.html	NT-Groups
list-NT-People.html	NT-People
list-Org-Units.html	Org-Units
list-Organizations.html	Organization
list-People.html	People

See Also

Chapter 6, “Search Attributes, Filters, and Results,” describes how search result templates can be edited to modify the display of search results.

Banner Files

Banner files identified in Table 4.5 are used to specify the banner and button images that appear in gateway forms.

Table 4.5 Banner Files Controlling Appearance of Gateway Forms

Banner Files	Controls Appearance of Banner and Buttons in
maintitle.html	main form
authtitle.html	authentication form
csearchtitle.html	advanced search forms
newentrytitle.html	new entry forms
searchtitle.html	standard search form
display-*.html	view or edit entry forms

See Also

Chapter 7, “Customizing Graphics and Color,” describes how to alter the color schemes and images appearing on gateway forms.

Chapter

5

Entry Types and Object Class Attributes

This chapter describes how entry type formats—defined by object classes and their attributes—are controlled by parameters in the dsgw.conf file. Topics include:

- Entry Types (Object Classes)
- Mapping Locations and Entry Types
- Object Class Attributes in Template Files

Entry Types (Object Classes)

Parameters Controlling Entry Types

The functionality of entry types appearing on gateway forms are controlled by parameters stored in dsgw.conf, including:

- template parameter settings that determine the types of objects that can be created and the attributes supported for object classes.
- newtype parameter settings that determine the DN formats to be applied to new entries.

- location parameter settings that determine where in the directory new entries reside.

See Also

Parameters in the dsgw.conf file are described in Appendix A, ".conf Parameters."

template

The template parameter maps the gateway's HTML templates for entry types to the Directory Server's LDAP object classes.

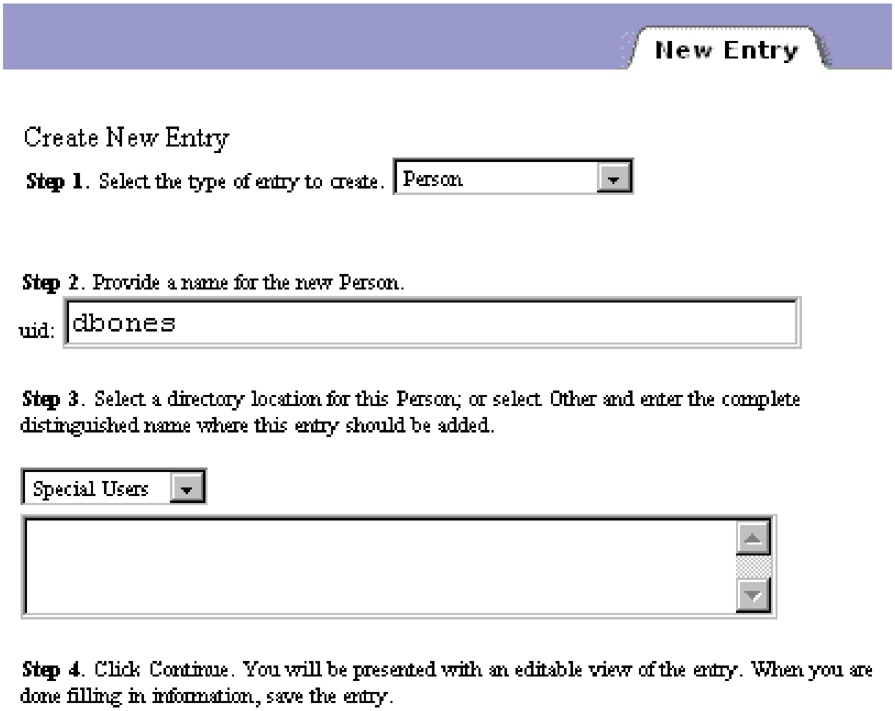
location

The location parameter defines points in the directory tree where new entries can be added. The location parameter definitions must precede newtype parameter definitions in the .conf file.

newtype

Each entry type is described by a newtype parameter. The newtype template indicates how new entries will be formatted and where in the directory tree entries new entries will appear. Entry types for the default gateway appear in the pull-down menu on gateway forms (for instance, the Create New Entry form, Figure 5.1).

Figure 5.1 New Entry Form



Create New Entry

Step 1. Select the type of entry to create. Person

Step 2. Provide a name for the new Person.

uid: dbones

Step 3. Select a directory location for this Person; or select Other and enter the complete distinguished name where this entry should be added.

Special Users

Step 4. Click Continue. You will be presented with an editable view of the entry. When you are done filling in information, save the entry.

Considerations for Adding New Entry Types

Before adding support for a new entry type (object class), decide where in the directory tree the entry type should appear and how the entry type should be formatted.

Deciding where the new entry appears

If a location parameter is not set up to point to the location where the new entry type appears, add a new location parameter to the dsgw.conf file.

Deciding how the new entry is formatted

When the new object class has many attributes in common with an existing entry type, update the corresponding template line in dsgw.conf to support the new object class. When a new object class requires a new template, add a new template parameter to dsgw.conf.

See Also

The newtype and location parameters are described in Appendix A, “.conf Parameters”

Mapping Locations and Entry Types

The location parameter is used to define points in the directory tree where new entries can be added. The default locations defined in dsgw.conf are intended for the sample directory shipped with the Directory Server. They are unlikely to match the structure of the actual directory.

Note For simple directory structures, define locations that represent branch points in the directory. For complex structures, define branch points for only the most commonly used directory branches.

Mapping Entry Types to Locations

Each Entry type must be mapped to a location where that type of entry can be placed. Example 5.1 shows a mapping of locations and newtype parameters in dsgw.conf.

Example 5.1 Mapping of Locations and Entry Types in dsgw.conf

```
location   country "United States"            "c=US#"
location   org     "This Organization"         ""
location   groups  “Organizational Groups"    "ou=Groups"
location   acct    "Accounting"               "ou=Accounting"
location   hr     "Human Resources"           "ou=Human Resources"
location   pay    "Payroll"                   "ou=Payroll"
location   pd     "Product Development"       "ou=Product Development"
location   test    "Product Testing"          "ou=Product Testing"
```

```
newtype   orgperson  "Person"            cn   acct hr pay pd test
newtype   NTperson  "NT Person"          cn   acct hr pay pd test
newtype   groupun   "Group"              cn   groups
newtype   orgunit   "Organizational Unit"ou   org
newtype   org       "Organization"       o    country
```

In Example 5.1, locations defined in the newtype parameter (boldfaced) correspond to handles defined in the location parameter (also boldfaced). The friendly names (in quotes) in the third column indicate the choices that appear in pull-down menus on gateway forms.

Note location parameter definitions in dsgw.conf must be listed before newtype parameter definitions.

See Also

"Entry Types (Object Classes)" on page 45

Setting Up Organizational Units

Assuming that the root DN is set to o=airius.com, the mappings in Example 5.1 can be used to create people in the following organizational units:

```
ou=Accounting, o=airius.com
ou=Human Resources, o=airius.com
ou=Payroll, o=airius.com
ou=Product Development, o=airius.com
ou=Product Testing, o=airius.com
```

Configuring DN Formats for Entry Types

UID-Based DN

When a person or NT person entry is added to the directory, the gateway prompts for a unique DN. The unique DN is typically the user ID of a person in the organization.

Although DN formats can be based on the common names of employees in the organization, common names are frequently not unique within an organization.

Note UID-based DN formats are recommended because they are by nature unique and can prevent naming collisions within the directory.

Modifying the Default DN format

The default DN format can be modified by editing the rdnattr variable within the newtype parameter.

To change the gateway configuration so that person entries are created using common name-based DNs rather than user ID-based DNs, edit the following line in the dsgw.conf file

```
newtype orgperson “Person” uid people special
```

to read

```
newtype orgperson “Person” cn people special
```

Object Class Attributes in Template Files

Default Gateway Object Classes

The default gateway supports the object classes listed in Table 5.1.

Table 5.1 Default Object Classes

Default Gateway Object Class	Related Template File
groupOfNames	display-group.html
groupOfUniqueNames	display-groupun.html
ntGroup	display-ntgroup.html
organization	display-org.html
organizationalUnit	display-orgunit.html
person	display-person.html
inetOrgPerson	display-orgperson.html
ntUser	display-ntuser.html
country	display-country.html

Templates and Directives

Object class attributes associated with an entry type are defined by *directives* contained in gateway template files. Directives are instructions, written as HTML comments, that are interpreted by the gateway's CGI scripts. Each directive is an independent, single line of HTML in a template file (with the exception of <!-- GCONTEXT -->, which is embedded within an URL).

Entry-Related Directives

Entry-related directives are responsible for how the gateway displays, edits, adds, and lists directory entries. The most commonly used entry-related directive is DS_ATTRIBUTE, which determines how attributes in LDAP entries are displayed on gateway forms.

DS_ATTRIBUTE directives begin with a DS_ENTRYBEGIN tag and close with a DS_ENTRYEND tag.

See Also

Appendix B, "Gateway Directives Reference" lists the possible arguments for the DS=ATTRIBUTE directive.

Adding Attributes to Object Classes

Adding an attribute to an object class requires adding an additional row to the HTML table in the template file where the object class is defined.

The syntax in Example 5.2defines an Initials attribute for the orgperson object class.

Example 5.2 Adding Initials attribute to orgperson object class

```
<TR>
<TD VALIGN= ”TOP” NOWRAP>Initials</TD>

<TD VALIGN= ”TOP” NOWRAP><B>
<!-- DS ATTRIBUTE “attr=Initials“ “syntax=cis“ “cols=>16“ -->
</B></TD>
<TD>&nbsp;</TD>
<TD>&nbsp;</TD>
```

```
</TR>
```

To complete the row, two null cells are added. This maintains the HTML table format. For Asian character sets, substitute an ideograhpic space for the non-breaking space (&nbsp;) shown in Example 5.2

Note Attribute values are added in pairs. When adding a single attribute to an object class, remember to complete the table row.

Deleting Attributes from Object Classes

Deleting an attribute from an object class requires deleting a complete row or part of a row from the HTML table where the object is defined. Example 5.3 shows the steps required to delete the mobile phone attribute from the orgperson object class.

Example 5.3 Deleting mobile phone attribute from orgperson object class

1. Open display-orgperson.html template and delete the mobile phone attribute-value pair (boldfaced text below)

   ```
   <TR>
   <TD VALIGN=”TOP” NOWRAP>Pager:</TD>
   <TD VALIGN=”TOP” NOWRAP><B>
   <!-- DS ATTRIBUTE “attr=pager” “syntax=tel” “cols=>16”
   -->
   </B></TD>
   <TD VALIGN=”TOP” NOWRAP>Mobile Phone:</TD>
   <TD VALIGN=”TOP” NOWRAP><B>
   <!-- DS ATTRIBUTE “attr=mobile” “syntax=tel” “cols=>16”
   -->
   </B></TD>
   </TR>
   ```

2. Insert a null pair to replace the deleted attribute (boldfaced below).

   ```
   <TR>
   <TD VALIGN=”TOP” NOWRAP>Pager:</TD>
   <TD VALIGN=”TOP” NOWRAP><B>
   <!-- DS ATTRIBUTE “attr=pager” “syntax=tel” “cols=>16”
   -->
   </B></TD>
   ```

```
<TD>&nbsp;</TD>
<TD>&nbsp;</TD>
</TR>
```

Note When deleting a single attribute-value pair from a row, replace the two deleted cells with two cells containing the non-breaking space character. This maintains the table width and ensures that the background colors are rendered correctly.

Extending Object Classes

The gateway can be extended to support additional object classes. This requires changing information in an existing object class template so that the gateway displays the associated entry type.

Note When extending object class definitions, the child should appear below the parent object class in the HTML file. Otherwise, the gateway cannot correctly interpret the HTML syntax.

Adding a Template for a Child of a Parent Class

The easiest way to create a new object class is to extend an existing object class template, adding and deleting attributes as necessary. Example 5.4 shows the steps required to add a template for a new object class, airiusPerson, which adds two custom attributes—dateOfBirth and preferredOS—to the inetOrgPerson object class.

Example 5.4 Adding template for subclass airiusPerson

1. Copy the display-orgperson.html file and rename it as display-airiusperson.html.
2. Edit the third line in the template file to indicate the name of the new directory entry type. Change:

   ```
   <!-- inet. organizational person directory entry -->
   ```

   to

   ```
   <!-- airius person directory entry -->
   ```

3. Edit the DS_OBJECTCLASS directive to include the new object class. Change:

```
<!-- DS_OBJECTCLASS "value=person,inetOrgPerson" -->
```

to

```
<!-- DS_OBJECTCLASS "value=person,inetOrgPerson,
airiusperson" -->
```

4. Add the value for airiusperson to the input type.

```
<!-- IF "Adding" -->
<INPUT TYPE="hidden" NAME="add_objectClass"
VALUE="top">
<INPUT TYPE="hidden" NAME="add_objectClass"
VALUE="person">
<INPUT TYPE="hidden" NAME="add_objectClass"
VALUE="organizationalPerson">
<INPUT TYPE="hidden" NAME="add_objectClass"
VALUE="inetOrgPerson">
<INPUT TYPE=”hidden” NAME=”add_objectClass”
VALUE=”airiusperson”>
```

Be sure to add the line immediately before the ENDIF directive:

```
<!-- ENDIF // Adding -->
```

5. Add a new table row containing the dateOfBirth and preferredOS attribute-value pairs.

```
<TR>
<TD VALIGN="TOP">Date of Birth:</TD>
<TD VALIGN="TOP"><B>
<!-- DS_ATTRIBUTE "attr=dateOfBirth" "cols=>8" -->
</B></TD>
<TD VALIGN="TOP">Preferred OS</TD>
<TD VALIGN="TOP"><B>
<!-- DS_ATTRIBUTE "attr=preferredOS" "cols=>6" -->
</B></TD></TR>
```

For more information on adding attributes, see “Object Class Attributes in Template Files” on page 50.

6. Define a template parameter in dsgw.conf for the object class airiusPerson:

```
template Airiusperson person inetorgperson
airiusperson
```

This instructs the gateway to display the Airiusperson entry type according to the template defined for the airiusPerson object class (display-airiusperson.html).

7. Update the Directory Server schema to include the airiusPerson object class.
8. To allow users to add entries for Airiusperson using the gateway, add an additional newtype parameter to the dsgw.conf file. If this entry type is for display purposes only, no newtype parameter needs to be added.

See Also

See "Considerations for Adding New Entry Types" on page 47, and "Extending Search Preferences" on page 64.

Creating a New Parent Object Class

These steps are required when the object class is not a child of an existing object class.

1. Add a template parameter to dsgw.conf for the new object class.

   ```
   template <newobjectclass>
   ```

   This instructs the gateway to display the associated entry type according to the template defined for the new object class.
2. To allow gateway users to add entries for the entry type, add an additional newtype parameter to the dsgw.conf file. If the associated entry type is for display purposes only, no newtype parameter needs to be added.
3. Update the Directory Server schema to include the new object class.
4. Add a search object entry to dsgwsearchprefs.conf and update dsgwfilter.conf so that the gateway searches for entries of this type.
5. Create a new search results form defining how the gateway displays search results for the new object class.

Note Netscape recommends modifying an existing search result form to create a new search results form.

See Also

See "Adding Information to Search Results" on page 69 and "Removing Information from Search Results" on page 70.

Chapter

6

Search Attributes, Filters, and Results

This chapter describes the files that control how the gateway searches for objects and how to add search support for a new object. Topics include:

- Search Configuration Files
- Changing Search Scope
- Modifying Search Attributes for Advanced Searches
- Adding Search Support for a New Object
- Modifying Default Search Filters
- Customizing Search Results Templates

Search Configuration Files

The search configuration files that control the gateway's search functionality are dsgwsearchprefs.conf and dsgwfilter.conf. These files are stored in the gateway's template directory (/config for the default gateway).

dsgwsearchprefs.conf

Preferences for searching object classes defined in the gateway are specified in dsgwsearchprefs.conf. Each entry contains

- the scope of the search within the Directory Server
- the search filter to append to the search string (corresponding to the search filter entry defined in dsgwfilter.conf)
- the label of the search attribute as it is displayed in the Find drop-down list on the search form
- the object class attribute to search on
- match types to use in search results

Note Define new search preferences in dsgwsearchprefs.conf whenever a new object class with searchable attributes is added to the gateway.

dsgwfilter.conf

dsgwfilter.conf contains an entry for each search object defined in dsgwsearchprefs.conf. Each entry defines:

- the pattern to search for
- the delimiters for the search pattern
- the LDAP filter for generating search results
- the text to use in describing search results for the pattern
- the scope of the search (not required)

The name of the search filter entry for a search object is identified in dsgwsearchprefs.conf.

Changing Search Scope

Search object entries in dsgwsearchprefs.conf define the search scope used in searches for the corresponding object class. The default scope for gateway search objects, subtree, specifies the baseurl and all its children.

The scope of a search object can be changed by editing the corresponding line in dsgwsearchprefs.conf. Valid search scopes are shown in Table 6.1.

Table 6.1 Valid Search Scopes

Search Scope	Tells the Gateway to
base	Search the Directory Server for the entry specified in the baseurl parameter but do not search in children of the entry.
onelevel	Do not search in the entry specified in baseurl, but search in the most immediate children of the entry.
subtree	Search the entry specified in the baseurl parameter and all of it's children. This is the default setting.

Modifying Search Attributes for Advanced Searches

Each search object entry in dsgwsearchprefs.conf has a list of attributes that can be modified for advanced searches.

Standard and Advanced Searches

An advanced search differs from a standard search in that users are provided with a pull-down menu of search types. In the default gateway, the standard search form is used to search on object classes defined for the gateway. The advanced search form allows users to search object class attributes and to specify matching patterns (Figure 6.1 shows the entry types that can be selected, Figure 6.2 shows the specific attributes that can be searched on for person entries, and Figure 6.3 shows the matching types that can be specified).

Figure 6.1 Entry Type Pull-Down Menu for Advanced Search Form

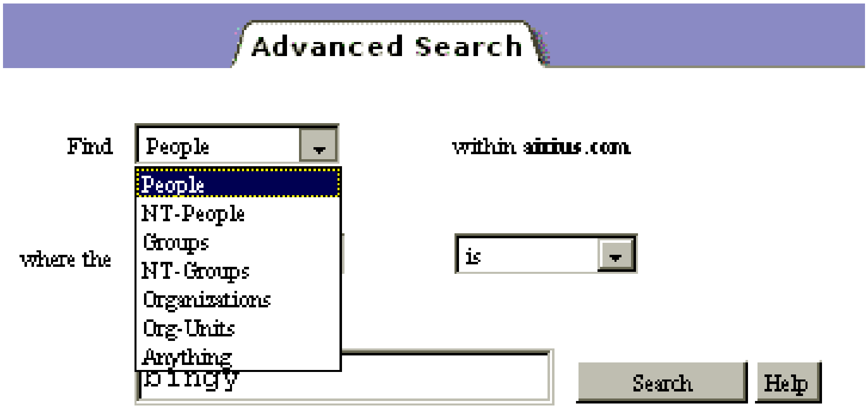


Figure 6.2 Attribute Pull-Down Menu on Advanced Search Form

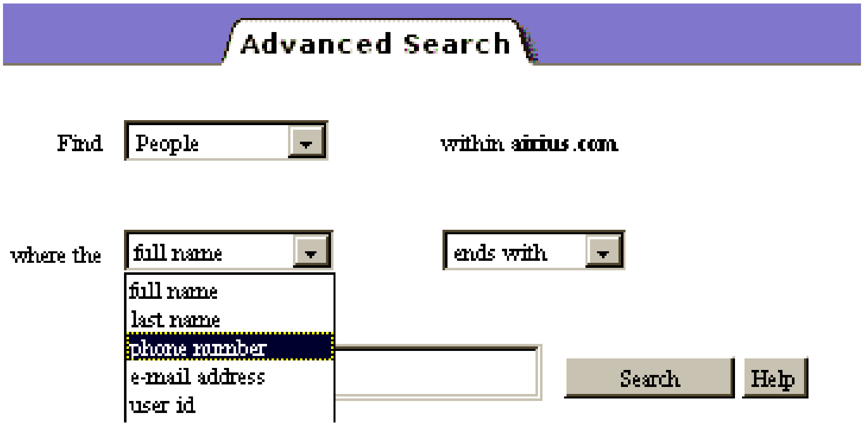
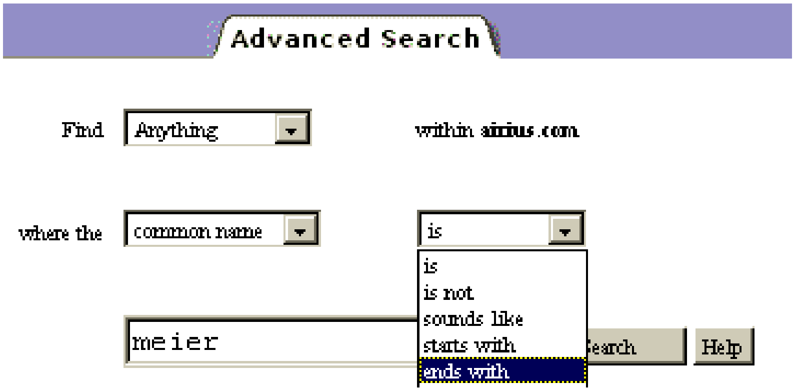


Figure 6.3 Matching Type Pull-Down on Advanced Search Form



Find Anything within airius.com

where the common name is

meier

Specifying Search Attributes for Person

The dsgwsearchprefs.conf syntax in Example 6.1 specifies that the cn, sn, telephoneNumber, mail, uid, and title attributes are used in a search for person entries.

Example 6.1 Search Attributes for Person Entry

```
People
“”
“Search for”:
“(&(objectClass=person)”
“dsgw-people”
subtree
"full name"        cn                 111111   ""    ""
"last name"        sn                 111111   ""    ""
"phone number"   "telephoneNumber"  111011   ""    ""
"e-mail address" "mail"             111111   ""    ""
"user id"        "uid"              111111   ""    ""
"title"           title             111111   ""    ""
END
"is"            "(%a=%v))"
"is not"        "(!(%a=%v)))"
"sounds like"   "(%a~=%v))"
"starts with"   "(%a=%v*))"
"ends with"     "(%a=*%v))"
"contains"      "(%a=*%v*))"
END
```

The first column in Example 6.1 specifies how the LDAP attribute shown in the second column appears in the pull-down menu on the Advanced Search form. Choose one term form menu, or leave out.

The third column contains a string of six bits. Each bit position in the string maps to a match type, as shown in Table 6.2. A value of 1 in the bit position indicates that the match type is valid for the associated attribute. A value of 0 indicates that the match type is not valid. In Example 6.1, the bit position for the telephone number attribute is set to 0, so the Directory Server does not search for sounds like match types for phone number entries on the Advanced Search form.

Table 6.2 Bit Positions and Corresponding Search Match Types

Bit position	Match type	Type of Search Performed
1	contains	Substring search for entries having attributes with values ending wih the specified search sting.
2	ends with	Substing search for entries having attribgutes with values ending with the sspecified search string.
3	starts with	Substing search for entries having attributes with values starting with the specified search string.
4	sounds like	Approximate search for phonetic matches.
5	is not	Equality search for all the entries with attribute value that does not match exactly.
6	is	Equality search for all entries having value that matches exactly.

The fourth and fifth columns in the search attributes contain empty strings required by the gateway. These should not be altered.

Directory Express Search Support for User ID

Directory Express looks for matches for user ID strings. It does not attempt to match user ID substrings.

To configure substring matching for user IDs, substring index the uid attribute and uncomment the appropriate lines in pbconfig/dsgwfilter.conf.

Adding Search Support for Additional Attributes

The boldface syntax in Example 6.2 shows the addition of preferences for searching the pagerTelephoneNumber attribute.

Example 6.2 Extending Search Preferences to Include Pager Number

```
People
“”
“Search for”:
“(&(objectClass=person)
“dsgw-people”
subtree
"full name"          cn                      111111    ""    ""
"last name"          sn                      111111    ""    ""
"phone number"      "telephoneNumber"        111011    ""    ""
"e-mail address"    "mail"                   111111    ""    ""
"user id"           "uid"                    111111    ""    ""
"title"              title                   111111    ""    ""
“pager number”      pagerTelephoneNumber    111011    ""    ""
```

As a result of adding this syntax to dsgwsearchprefs.conf:

- A pager number selection appears in the pull-down menu on the Advanced Search form.
- The gateway searches the pagerTelephoneNumber attribute of all entries within the scope of the search.
- The gateway looks for values that contain, end with, start with, or are identical to the search string entered by the user. It does not look for values that sound like the search string entered by the user.

Adding Search Support for a New Object

There are two ways to add search support for a new object:

- Update entries in dsgwsearchprefs.conf and dsgwfilter.conf with definitions of search attributes for the new object. Use this method to add search support for an object that is a child of another object.
- Create new entries in dsgwsearchprefs.conf and desgwfilter.conf for a new object class. This method requires setting preferences for searching object class attributes and defining a filter to use in expressing search results.

Extending Search Preferences

The boldface syntax in Example 6.3 shows the introduction of a new object, airiusPerson, and a new attribute, dateOfBirth, to the search preferences for the person object class.

Example 6.3 Editing dsgwsearchprefs.conf to Extend a Search Preference

```
People
““
“Search for”:
“(&(objectClass=person)(objectClass=airiusPerson)”
“dsgw-people”
subtree
"full name"            cn                111111    ""    ""
"last name"            sn                111111    ""    ""
"phone number"         "telephoneNumber" 111011    ""    ""
"e-mail address"       "mail"            111111    ""    ""
"user id"              "uid"             111111    ""    ""
"title"                title             111111    ""    ""
“birthdate”            dateOfBirth       111011    “ “    “ “
```

As a result of adding the airiusPerson object class in Example 6.3:

- The gateway includes airiusPerson entries in its search for persons.
- A “birthdate” field appears on the advanced search form.
- The gateway searches the dateofBirth attribute of all entries within the scope of the search.

The syntax added to dsgwfilter.conf in Example 6.4 defines the search values for the dateOfBirth attribute.

Example 6.4 dsgw-people entry configured to search for dateofBirth values

```
“[0-9][0-9]/[0-9][0-9]/[0-9]0-9]” “ “ “
    (dateOfBirth=%v))”     “date of birth is”
    (dateOfBirth=%v*))”    “date of birth starts with”
```

Creating a New Search Object

To add a new search object that differs from existing search objects:

- add a new entry in dsgwsearchprefs.conf
- add a new entry in dsgwfilter.conf
- add a new search results template

The syntax in Example 6.5 specifies that the search definition for OrgRole searches the cn, l, roleOccupant, description, and telephoneNumber attributes.

Example 6.5 Creating a new dsgwsearchprefs.conf entry for OrgRole

```
“Org-Roles”
""
"Search For:"
"(&(objectClass=organizationalRole)
"dsgw-orgrole"
not-used-by-dsgw
not-used-by-dsgw
subtree
"name"            “cn”                111111    ""    ""
"location"        "l"                 111111    ""    ""
"occupant"        "roleOccupant"      111111    ""    ""
"description"     “description”       111011    ""    ""
"phone number"    "telephoneNumber"   111011    ""    ""
END
"is"            “(%a=%v))"
"is not"        “(!(%a=%v)))"
"sounds like"   “(%a~=%v))"
"starts with"   “(%a=%v*))"
"ends with"     “(%a=*%v))"
"contains"      “(%a=*%v*))"
END
```

The syntax in Example 6.6 specifies a list of filters for searching attributes in the orgRole object class.

Example 6.6 Creating a new dsgwfilter.conf entry for orgrole

```
dsgwfilter.conf:
dsgw-orgrole"
"="       " "   "(%v))"    "LDAP filter is"

"^[+]*[0-9][ 0-9-]*$" " "   "(telephoneNumber=*%v))" "phone number ends
with"
"^\*$"   "  "     "(cn=*))" "name is"
".*"   ". _" "(cn=%v1-))"  "name is"
"(cn=*%v1-*))"            "name contains"
"(cn~=%v1-))"             "name sounds like"
```

See Also

Modifying search results forms is described in “Modifying Search Result Templates” on page 69

Modifying Default Search Filters

The gateway uses dsgwflter.conf to map patterns in search strings to a relevant search filter and search result description (a search pattern is a grep-style regular expression). dsgwwfilter.conf can be optimized to respond to common user data patterns.

Netscape recommends modifying existing search filters in dsgwfilter.conf to support new user data patterns, instead of creating new filters.

Search Filters for User Data Patterns

Example 6.7 shows typical search filter syntax for any search string containing the at (@) symbol. In this example, the gateway responds to search strings containing the @ symbol (the pattern) by searching the mail attribute for values that are equal to or start with the supplied value (the filter). The gateway returns a message on the search results form indicating the number of entries where the “email address is” or “email address starts with” the user-supplied value (the description).

Example 6.7 Search Filter for Email Address Searches

```
"@"    " ""(mail=%v))"    "email address is"
    "(mail=%v*))"    "email address starts with"
```

Note Standard searches use the filters associated with the first matching pattern only. Advanced searches use all filters defined for the entry.

Specifying a Search Filter for a New Object

The syntax in Example 6.8 allows users to search person entries by birthday or birth month.

Example 6.8 Modifying a search filter pattern

```
[0-9][0-9]/[0-9][0-9]/[0-9]0-9](dateOfBirth=%v))date of birth is
dateOfBirth=%v*))date of birth starts with
```

As a result of the adding the boldface syntax in Example 6.8 to the dsgw-people entry in dsgwfilter.conf, the gateway also filters the dateofBirth attribute for values that start with the supplied value (the filter). The gateway returns a message on the search results form indicating the number of entries where the "date of birth is" or "date of birth starts with" the user-supplied value (the description).

Note Make sure to place new patterns near the top of the pattern definitions for a given object. For example, in the dsgw-people entry, place customized patterns before the pattern that begins with the @ symbol. Patterns near the end of the entry are more general, and match many different strings.

Customizing Search Result Templates

How the Gateway Displays Search Results

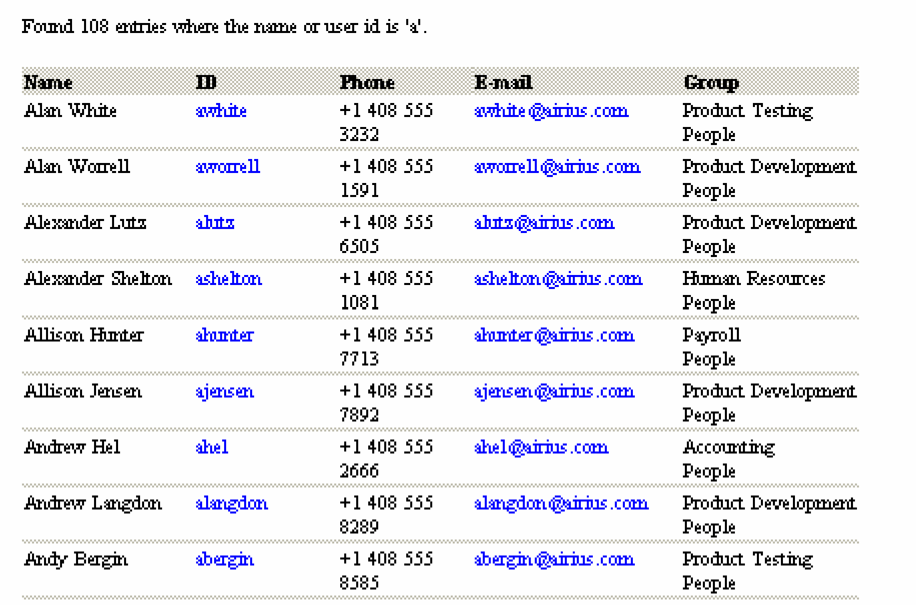
When a user submits a standard search or advanced search from the gateway, the gateway constructs a search string and filter for the corresponding search object and queries the Directory Server. The Directory Server responds with matching entries in the LDAP database. The gateway uses a search result template to display the entries returned by the Directory Server.

Search Result Tables

Search results are displayed as tabular data. Headings in each result tables reflect the object attributes identified in the search result template.

For example, the heading row on the search results form for a People search displays the Name, Phone Number, E-Mail Address, and Title attributes.

Figure 6.4 Search Results



Found 108 entries where the name or user id is 'a'.

Name	ID	Phone	E-mail	Group
Alan White	awhite	+1 408 555 3232	awhite@airius.com	Product Testing People
Alan Worrell	aworrell	+1 408 555 1591	aworrell@airius.com	Product Development People
Alexander Lutz	alutz	+1 408 555 6505	alutz@airius.com	Product Development People
Alexander Shelton	ashelton	+1 408 555 1081	ashelton@airius.com	Human Resources People
Allison Hunter	ahunter	+1 408 555 7713	ahunter@airius.com	Payroll People
Allison Jensen	ajensen	+1 408 555 7892	ajensen@airius.com	Product Development People
Andrew Hel	ahel	+1 408 555 2666	ahel@airius.com	Accounting People
Andrew Langdon	alangdon	+1 408 555 8289	alangdon@airius.com	Product Development People
Andy Bergin	abergin	+1 408 555 8585	abergin@airius.com	Product Testing People

Table 6.3 lists the default gateway search objects and the information displayed on the search results list. Search results templates are stored in the dsgw/config directory and use the list-<search object>.html file naming convention.

Table 6.3 Default Search Results for Search Objects

Search object	Search result template used	Search results displayed
people	list-People.html	name, phone number, e-mail address, and title
NT-people	list-NT-People.html	name, NT domain, NT username, and phone number
Groups	list-Groups.html	group name and description
NT-Groups	list-NT-Groups.html	LDAP group name, NT domain name, NT group name, and description
Organizations	list-Organizations.html	organization name, description, and phone number
Org-Units	list-Org-Units.html	organizational unit name, description, and phone number
Anything	list-Anything.html	name, phone number, e-mail address, and description

A new search result template is required for each new object class that is not a child of another object class.

Modifying Search Result Templates

Adding Information to Search Results

The boldface text in Example 6.9 shows how additional information can be added to the search result by modifying the corresponding search result template.

Example 6.9 Adding OrgUnit to Search Results for People

```
<TR>
```

```
<TH NOWRAP>Name<TH NOWRAP>Phone Number<TH NOWRAP>E-Mail Address<TH
NOWRAP>Title<TH NOWRAP>Organizational Unit
</TR>

<TR>
<TD NOWRAP>
<!-- DS_ATTRIBUTE "attr=dn" "syntax=dn" "label=" -->
onMouseOver="window.status='Click here to view this entry in detail';
return true">
<!-- DS_ATTRIBUTE "attr=cn" "numfields=0" "defaultvalue=name"
"options=readonly" -->
</A>
<TD NOWRAP>
<!-- DS_ATTRIBUTE "attr=telephoneNumber" "syntax=tel" -->
<TD NOWRAP>
<!-- DS_ATTRIBUTE "attr=mail" "syntax=mail" -->
<TD NOWRAP>
<!-- DS_ATTRIBUTE "attr=title" --></TD>
<TD NOWRAP>
<!-- DS_ATTRIBUTE “attr=ou” “syntax=cis” --></TD>
</TR>
```

In Example 6.9, the additional HTML table heading syntax adds the Organizational Unit label to the heading row of the table. The additional DS_ATTRIBUTE directive syntax adds a cell to the body row of the table indicating that the information is stored in the ou attribute of the entry and the string is case insensitive.

Removing Information from Search Results

To remove information from a search result, remove the tag that creates the table head cell labeling the attribute and the tag that creates the Directory Server call for the corresponding attribute value from the corresponding list-<search object>.html file.

For example, to remove the NT Domain attribute from the list-NT-People.html search results file, delete the <TH NOWRAP>NT Domain tag from table heading. The table cell containing the <!-- DS_ATTRIBUTE “attr=ntuserdomainid” “syntax=ntdomain” --> directive would also need to be removed.

Chapter

7

Customizing Graphics and Color

This chapter describes how to change the appearance of default gateway forms. Topics include:

- Updating the Banner Image
- Updating Button Images
- Color Schemes
- Changing Table Colors

Appearance of Gateway Forms

The default gateway installed during Directory Server installation matches the standard appearance of Netscape products. The gateway Interface Templates can be modified to customize the appearance of:

- the banner image
- buttons
- background color
- text color

- link color

Updating the Banner Image

The default gateway banner image that appears at the top of the gateway forms is title.gif. This image can be replaced by a different banner image stored as dsgw/html/title.gif.

Changing Dimensions of Banner Image

The default Netscape banner image has a height of 40 pixels and a width of 530 pixels. When using a banner image with a different pixel height and width, change the specifications of the image in all files in dsgw/html where the image is referenced:

- maintitle.html
- authtitle.html
- searchtitle.html
- csearchtitle.html
- newentrytitle.html

Changing Filename of Banner Image

Netscape recommends maintaining the default filename—title.gif—for the banner image. Changing the default filename of the banner image requires updating the filename in all files where the image is referenced.

Note Any image used to replace title.gif must be a true gif image. Images in other formats (PICT, EPS, BPX, TIFF) will not appear as intended.

Updating Button Images

Buttons on gateway forms can be replaced by updating button image files stored in the dsgw/html directory. Table x describes the button image files stored in the dsgw/html directory:

Table 7.1 Button Images

Button Image	Description
stsearch_off.gif	used in the maintitle.html, authtitle.html, csearchtitle.html, and newentrytitle.html pages
stsearch_on.gif	used on the searchtitle.html page
adsearch_off.gif	used in the maintitle.html, authtitle.html, searchtitle.html, and newentrytitle.html pages
adsearch_on.gif	used on the csearchtitle.html page
newentry_off.gif	used in the maintitle.html, authtitle.html, csearchtitle.html, and searchtitle.html pages
newentry_on.gif	used on the newentrytitle.html page
authen_off.gif	used in the maintitle.html, searchtitle.html, csearchtitle.html, and newentrytitle.html pages
authen_on.gif	used on the authtitle.html page

The default Netscape button images have a height of 24 pixels and a width of 122 pixels. If the new button image uses a different pixel height and width, these specifications must be changed in all files in dsgw/html where the image is referenced:

- maintitle.html
- authtitle.html
- searchtitle.html
- csearchtitle.html

- newentrytitle.html

Changing the default filename of a button .gif file requires updating the filename in all files where the image is referenced.

Any image used to replace a button image must be a true gif image. Images in other formats (PICT, EPS, BPX, TIFF) will not appear as intended.

Color Schemes

Changing the color schemes for a form requires editing the files that make up a gateway form. The procedure for changing colors depends on the gateway file type.

Files Controlling Colors on Gateway Forms

Table 7.2 describes the gateway files that control the appearance of gateway forms. These files may need to be updated when changing the appearance of the banner image, button images, or background and body colors.

Table 7.2 Files Controlling Appearance of Gateway Forms

To Change Colors on the	Edit	File Type
authentication forms	authtitle.html	banner
	authPassword.html	script
	authSearch.html	script
standard search forms	searchtitle.html	banner
	searchString.html	script
	greeting.html	HTML
	list-*.html	search result
advanced search forms	csearchtitle.html	banner
	csearchAttr.html	script
	csearchBase.html	script
	csearchMatch.html	script
	csearchString.html	script
	csearchType.html	script
	emptyFrame.html	HTML
	list-*.html	search result
new entry forms	newentrytitle.html	banner
	newentryType.html	script
	newentryName.html	script
	display-*.html	template
view or edit entry forms	display-*.html	template

Changing Colors Using BODY Tag

Table 7.3 lists the five standard color attributes that can be changed.

Table 7.3 Colors and Related Attributes for BODY tag

Color	Using Attribute
text color	TEXT
background color	BGCOLOR
link color	LINK
visited link color	VLINK
selected link color	ALINK

To adjust background colors in the banner files and other standard HTML files, edit the HTML to change the color associated with the BGCOLOR and TEXT attributes to a different RGB color value (for example, to change background to aquamarine and text to white, use <BODY BGCOLOR="#70DB93" TEXT="#9932CD">)

Changing Colors Using Directives

To change background colors in script files, add a COLORS directive immediately before the BODY tag. The syntax in Example 7.1 shows the use of the COLORS directive to change text color to blue, background color to aqua, and the link color to red.

Example 7.1 Changing color settings using COLOR directives

```
<!-- COLORS “attribute=color [attribute=color [...]]” -->

<!-- COLORS “TEXT=#0000FF BGCOLOR=#00FFFF LINK=#FF0000” >

DS_NEWENTRY_TYPE_BODY -->
```

In Example 7.1, attribute is a standard HTML %color attribute and color is an RGB color value in the form #rrggbb (or a standard color name, for example, aquamarine).

Changing Table Colors

Specifying Color Names and Color Codes

There are two ways to specify colors:

- Use a color value, a six-digit hexadecimal number known as the red-green-blue (RGB) triplet. The RGB triplet always begins with a hash mark (#) followed by 3 2-digit codes that represent the amount of red, green, and blue that make up the color (#rrggbb). There are over 16 million RGB color codes.
- Use a color name. There are sixteen standard case-insensitive color names. Table 7.4 lists the sixteen standard color names and their equivalent RGB values

Table 7.4 Sixteen Standard Colors and their Hexadecimal Values

Color Name	Hexadecimal Value
black	#000000
silver	#C0C0C0
gray	#808080
white	#FFFFFF
maroon	#800000
red	#FF0000
purple	#80080
fuchsia	#FF00FF
green	#008000
lime	#00FF00
olive	#808000

Table 7.4 Sixteen Standard Colors and their Hexadecimal Values

Color Name	Hexadecimal Value
yellow	#FFFF00
navy	#000080
blue	#0000FF
teal	#008080
aqua	#00FFFF

Changing Color of Table Headings

To change the color of table heading text, edit the following lines within the template file, so that the BGCOLOR value is an RGB color value in the form #RRGGBB or a standard color name:

```
<TR BGCOLOR=006666 COLSPAN=4><FONT FACE=ARIAL, HELVETICA COLOR=WHITE>
```

The font color can be changed from white to another color using the same method. Within a single template file there may be multiple tables and consequently multiple table head rows that need to be modified to maintain a consistent color scheme.

Changing the Background Color of Table Rows

To change the color of the table body rows, edit the following line for each table within the template file so that the BGCOLOR attribute specifies the RGB color value in the form #rrggbb or a standard color name representing the new color:

```
<TABLE CELLSPACING=”2” BORDER BGCOLOR=#F2F2F2 ...
```

Appendix

A

.conf Parameters

This appendix describes, in alphabetical order, gateway configuration parameters. Parameters are defined in dsgw.conf and pb.conf, installed during Directory Server installation.

Associated directives are described in Appendix B, "Gateway Directives Reference."

authlifetime

Description Specifies the amount of time in seconds before a user's authentication expires in the gateway. When authenticating to the Directory from the gateway, the gateway retains authentication credentials for the amount of time specified in this parameter. Once authentication credentials have expired, the gateway prompts the user to re-authenticate.

Format

```
authlifetime <seconds>
```

Example The following example causes user authentication to expire in two hours. This is the default expiration time:

```
authlifetime 7200
```

baseurl

Description Specifies the host name and port number used to contact the Directory Server. This parameter also determines the search base used for searches performed from the gateway, and whether the gateway uses SSL to communicate with the Directory Server.

Format

```
baseurl [ldap | ldaps]://<host>:<port>/<search base>
```

ldap | ldaps . Use ldap to have the gateway communicate the Directory Server without using SSL. Use ldaps to have the gateway communicate with the Directory Server using SSL.

<host>. Indicates the host name of the device where the Directory Server is installed.

<port>. Indicates the port number used by the Directory Server. Always specify a port number even when using standard ldap or ldaps port numbers (389 and 636, respectively).

<search base> . Indicates the distinguished name representing the point in the directory from which all searches are performed. Normally, the search base is set to the directory's suffix.

Netscape recommends substituting the following hexadecimal values for the equal sign, space, and comma in the search base:

- use %3D instead of an equal sign (=)
- use %20 instead of a space ( )
- use %2C instead of a comma (,)

Example The following example sets the base URL to use SSL communications to a server running on the well-known LDAP security port (636). The base search address is set to o=airius.com:

```
baseurl "ldaps://dirserver.airius.com:636/o%3Dairius.com"
```

binddnfile

Description Specifies the location of the file where the bind DN and bind password are stored. This file is used to authenticate to the server for non-anonymous searching.

The binddnfile contains two lines, the first specifying the dn with which to bind, and the second specifying the bind password. For example:

```
binddn “<dn>”
bindpw <password>
```

This file should be stored separately of the .conf file for the gateway instance.

Format `binddnfile <filename>`

Example `binddnfile /export/TEST/bindfile`

changeHTML

Description Used by the gateway to substitute ideographic space for nonbreaking space (&nbsp;) in Asian character sets.

Format `changeHTML <nbsp_from> <nbsp_to> <charset>`

Example `changeHTML <space character> <space character> Shift_JIS`

charset

Description Defines the default character set for communication with HTTP clients. The default is UTF-8 (Unicode), which supports all the characters in the Netscape Directory. UTF-8 is the preferred character set, however many browsers don’t support the UTF-8 charset, or display it poorly.

Some users may require a different character set than the one specified using this parameter. For these users, the charset parameter setting may be overridden by creating a <LANG>/dsgw/charset.conf file which contains the charset name. However, to receive the correct language, users will have to configure their browsers to send the appropriate accept-language headers.

For compatibility with HTTP clients that can't handle an HTTP response with a charset parameter in the content-type, comment out this parameter in the.conf file. this prevents the gateway from sending an explicit charset to gateway clients. When no charset parameter is defined, the gateway by default transmits ISO-8859-1 (Latin-1).

The charset parameter is ignored by Netscape Communicator 4.0 and Internet Explorer 4.0 and greater clients, which request the UTF-8 charset by default. Forcing these clients to use a non-UTF-8 charset (such as Latin-1) requires the ignoreAcceptCharsetFrom parameter, introduced in Directory Server 4.0.

Format

```
charset <charset>
```

Example

```
charset UTF-8
```

For more information about charsets, see RFC 1345, which defines the syntax of charset names.

configdir

Description Specifies the location of the configuration directory of the gateway. These include the object class templates, search configuration files, search result templates, and script files used to dynamically generate HTML forms for the user.

The configuration directory for the default gateway (dsgw.conf) is ../config. The configuration directory for Directory Express (pb.conf) is ../pbconfig.

Format

```
configdir "<configuration directory>"
```

Example

```
configdir "../airiusconfig"
```

dirmgr

Description Specifies the distinguished name of the directory manager. This is the DN used to bind to the Directory Server when users authenticate as the directory manager from the gateway. Netscape recommends using a DN other than the root DN for this purpose. It is intended that the DN specified here has read and write authority for the subtree that the gateway sees.

Format

```
dirmgr "<distinguished name>"
```

Example

```
dirmgr "cn=Directory Manager, o=airius.com"
```

For information on the root DN and on setting permissions for the directory, see the *Netscape Directory Server Administrator's Guide*.

gwnametrans

Description Used by gateway CGIs to specify the URL to output for HTTP redirection. This needs to be specified as "/dsgw/<htmldir>" and should be the same as the NameTrans set in the HTTP server, if any is being used.

Format

```
gwenametrans "<HTTP redirect>"
```

Example

```
gwnametrans "/dsgw/pbhtml/"
```

htmldir

Description Specifies the location of the HTML files for the gateway. These include the HTML files controlling the appearance of gateway forms.

The HTML directory for the default gateway (dsgw.conf) is ../html. The HTML directory for Directory Express (pb.conf) is ../pbhtml.

Format

```
htmldir "<html directory>"
```

Example

```
htmldir "/airiusconfig"
```

ignoreAccetpCharsetFrom

Description Ignores request headers for the UTF-8 character set automatically sent by Netscape Communicator 4.x and Internet Explorer 4.x browsers. Can be used together with the charset parameter to transmit a charset other than Unicode to all gateway clients.

Format

```
ignoreAcceptCharsetFrom <HTTP client version string>
```

Example

```
ignoreAcceptCharsetFrom Mozilla/4.01x-NSCP Mozilla/3
```

include

Description Specifies the location of another config file that should be read by the gateway.

Format

```
include "<config file>"
```

Example

```
include "../config/dsgw-l10n.conf"
```

location

Description Defines the location choices for adding new entries to the gateway. Each location parameter represents a branch point in the directory tree below which new entries can be added.

Format

```
location <handle> "<friendly name>" "<dn>"
```

<handle>. An arbitrary string used by the location-suffix parameter to map a type of entry to the locations where the entry can be created.

<friendly name> . An arbitrary string that represents the location. This string should describe the location because the gateway displays this string to users to represent the location.

<dn>. The distinguished name representing this branch point in the directory. If this value is not terminated with a pound sign, the value specified on the include parameter is appended to this value to build the fully qualified distinguished name. If dn is terminated with a hash mark (#), the value represented here is assumed to be a fully qualified distinguished name, and the pound sign is stripped from the distinguished name before the DN is used by the gateway.

Example The following example defines an entry creation location in a user directory. This location corresponds to the Marketing organizational unit, and the remainder of the distinguished name is built from the value set in the include parameter:

```
location marketing "Marketing Organization" "ou=Marketing"
```

A slightly different example defines the same location but specifies the fully qualified distinguished name:

```
location marketing "Marketing Organization" "ou=Marketing,
o=airius.com#"
```

location-suffix

Description Identifies the directory suffix used to create new entries in the directory.

This value is appended to the DN field of the NLS parameter when the gateway is used to create new entries in the directory.

Format

```
location-suffix “<suffix>”
```

Example

```
location-suffix “o=airius.com”
```

newtype

Description Defines the types of entries that can be added to the directory using the gateway. The newtype parameter also defines the locations in the directory where an entry type can be added. For a user to create the entry, the corresponding location must be defined using the location parameter.

Format

```
newtype <template_name> “<friendly_name>” <rdnattr> <locations>
```

<template_name> . The name of a display-<template_name>.html file that defines the object class listed. Template files are stored in the ../config directory. The gateway uses these files to define how various types of entries are displayed when entries are being created or viewed:

- orgperson—corresponds to the display-orgperson.html template. Defines how the gateway displays an entry of object class type inetOrgPerson.
- groupun—corresponds to the display-groupun.html template. Defines how the gateway displays an entry of object class type groupOfUniqueNames.
- orgunit—corresponds to the display-orgunit.html template. Defines how the gateway displays an entry of object class type organizationalUnit.
- org—corresponds to the display-org.html template. Defines how the gateway displays an entry of object class type organization.

<friendly_name>. An arbitrary string that describes the entry. This string should be reasonably descriptive of the entry type because the gateway displays this string to users who are adding entries.

<rdnattr>. The attribute used to name entries of this type. For example, the default value for the rdnattr field for people entries is uid. This means that any people entries created using the gateway will have DNs of the following format:

```
uid=<string>
```

The rdnattr field can be modified so that entries are named using a different attribute. For example, to change the rdnattr of the newtype orgperson line from uid to cn, people entries created using the gateway will have cn-based DNs rather than the UID-based DNs (the default setting).

<locations>. A space-separated list of the locations where this type of entry can be added. The locations in this list must be identical to the <handle> specified on the corresponding location parameter.

Example The following example allows persons to be added to the Marketing subtree using the template for organizationalPerson:

```
newtype orgperson"Person"cn marketing
```

NLS

Description Identifies the libNLS data directory, which should contain a directory named "locales", containing the configuration files LANG.ctx, LANG.col, and LANG.txt for each supported language (locale).

Format

```
NLS <libNLS data directory>
```

Example

```
NLS ../../lib/nls
```

securitypath

Description Identifies the location of the certificate database used by the gateway when using SSL to communicate with the Directory Server. The certificate database contains the Certificate Authority issuing the certificate for the Directory Server. [rob--helpful to say that on install for DE and dsgw, this points to the certificate database for the admin server?]

Format

```
securitypath <NSHOME>/alias/<cert.db>
```

Example

```
securitypath "/export/TEST/alias/pb-cert.db"
```

template

Description Maps specific object classes to internal gateway templates. These templates define how a specific object class such as a person, a group, or an organizational unit is displayed in the gateway. The templatename identified has a corresponding HTML template stored in dsgw/conf/.

Format

```
template <templatename> <objectclass>
```

Example The following example identifies orgperson as the template defining attributes for person and inetorgperson object classes:

```
template orgperson person inetorgperson
```

vcard-property

Description The Directory Server gateway allows users to view vCards for person and NT person directory entries. The vCard and LDAP specifications define different labels to access information: vCards use properties and LDAP uses attributes. Therefore, there must be a way to map the vCard property names to the LDAP attribute names so that the Directory Server can locate the information for the vCard display. The vcard-property parameter accomplishes this vCard property to LDAP attribute mapping.

Format

```
vcard-property <vcardprop> <syntax> <ldapattr> [<ldapattr2>...]
```

<vcardprop>. The name of a vCard property. The following vCard properties are mapped to LDAP attributes:

- FN—The Formatted Name property. All vCards must have a FN property. By default, FN is mapped to the cn attribute.
- N—The Name property. By default, N is mapped to the sn and givenName attributes.
- ORG—The ORG property may refer to the organizational name and units of the person or resource associated with the vCard. By default, ORG is mapped to the o and ou attributes.
- ROLE—The ROLE property may refer to the role, occupation or business category of the person or resource described by the vCard. By default, ROLE is mapped to the businessCategory attribute.

- ADR;WORK—The work address of the of the person or resource described by the vCard. By default, ADR;WORK is mapped to the postalAddress attribute.
- ADR;HOME—The home address of the of the person or resource described by the vCard. By default, ADR;HOME is mapped to the homePostalAddress attribute.
- EMAIL;INTERNET—The email address of the person or resource described by the vCard. By default, EMAIL;INTERNET is mapped to the mail attribute.
- TITLE—The TITLE property specifies the job title, functional position or function of the person or resource described by the vCard. By default, TITLE is mapped to the title attribute.
- TEL;WORK—The business telephone number of the person or resource described by the vCard. By default, TEL;WORK is mapped to the telephoneNumber attribute.
- TEL;FAX—The fax number of the person or resource described by the vCard. By default, TEL;FAX is mapped to the facsimileTelephoneNumber attribute.
- TEL;CELL—The cellular telephone number of the person or resource described by the vCard. By default, TEL;CELL is mapped to the mobile attribute.
- TEL;HOME—The residential telephone number of the person or resource described by the vCard. By default, TEL;HOME is mapped to the homePhone attribute.
- NOTE—Provides any additional comments or information about the person or resource described by the vCard. By default, NOTE is mapped to the description attribute.

<syntax>. A string that describes the nature of the vCard information. The following syntaxes are supported:

- cis—used for simple strings, such as a person's name or telephone number
- mls—used for multiline strings, such as a mailing address

<ldapattr> [<ldapattr2>...]. The attribute(s) to be mapped to the vCard property. This is useful when mapping a vCard property to a custom attribute.

Example The following example changes the mapping of the NOTE property from the default description attribute to a custom attribute named hobbies:

```
vcard-property NOTE mls hobbies
```

Appendix B

Gateway Directives Reference

This appendix describes directives used in Gateway HTML object class and search result templates. Contents include:

- Introduction
- Context-Related Directives
- Entry-Related Directives
- Miscellaneous Directives

Introduction

The display of LDAP directory information is controlled by HTML template files containing directives. Directives are HTML comments that can be interpreted by the gateway CGIs.

The most commonly used directive is DS_ATTRIBUTE, used to display attributes present in LDAP entries. Here are some other examples of directives:

```
<!-- DS_HELPBUTTON "topic=HELP-ME-NOW" -->

<!-- DS_ATTRIBUTE "attr=sn" "size=>20" -->

<!-- IF "BoundAsThisEntry" -->

<!-- ENDIF -->
```

Note With the exception of GCONTEXT, each directive must start at the beginning of a line and be contained on a single line in the HTML file. Most of the directory server gateway directives begin with DS_, although some do not.

Structure of an HTML Template

Directory entry display, edit, and add templates generally have the following structure:

```
<HTML>
<HEAD>
<!-- DS_ENTRYBEGIN -->
<!-- DS_EMIT_BASE_HREF -->
<!-- BODY -->
<!-- DS_LAST_OP_INFO -->
<!-- DS_BEGIN_ENTRYFORM -->
<!-- attribute directives, e.g., -->
<!-- DS_ATTRIBUTE "attr=givenName" "size=>20" -->
<!-- DS_ATTRIBUTE "attr=sn" "size=>20" -->
<!-- etc. -->
<!-- DS_SAVEBUTTON "label= SAVE " -->
<!-- DS_END_ENTRYFORM -->
<!-- DS_ENTRYEND -->
<!-- ENDHTML -->
```

Structure of an HTML Template for Directory List

Directory entry list templates generally have the following structure:

```
<HTML>
<!-- TITLE "Search Results" -->
<!-- DS_SEARCHDESC -->
<!-- IF "FoundEntries" -->
<!-- DS_SORTENTRIES "attr=XXX" -->
<!-- DS_ENTRY_BEGIN -->
<!-- stuff that is repeated for each entry found, e.g., -->
<!-- DS_ATTRIBUTE "attr=dn" "syntax=dn" -->
<!-- etc. -->
<!-- DS_ENTRYEND -->
<!-- ELSE -->
<!-- stuff to be rendered if no entries were found, e.g.,-->
Please try a different search....
<!-- ENDIF -->
<!-- ENDHTML -->
```

Context-Related Directives

The context-related directives GCONTEXT and PCONTEXT appear within a line, and are not required to appear at the beginning of a line. This is an exception to the rule. All other directives must appear at the beginning of a line, to be recognized by the directory server.

GCONTEXT

Description

The <!-- GCONTEXT--> directive appears within an URL and is used in the invocation of CGIs through GET operations. <!-- GCONTEXT--> can appear anywhere on a line, and more than once within a line. The Gateway CGI reading <!--GCONTEXT --> replaces it with the Gateway context it has at the time.

Arguments

None.

Example

```
<a href=/dsgw/bin/lang?<?-- GCONTEXT -->&file=auth.html>click</a>
```

PCONTEXT

Description

The <!-- PCONTEXT--> directive must appear on a line by itself. The Gateway CGI reading <!--PCONTEXT --> replaces it with a hidden variable indicating the context it has at the time.

Arguments

None.

Example

```
<form method=post action=/dsgw.bin/dosearch>
```

```
<input type=hidden name=dn valute=””>
```

```
<!-- PCONTEXT -->

<form>
```

Entry-Related Directives

Entry-related directives are supported by the dosearch and edit CGIs.

DS_ENTRYBEGIN

Description

Delimits the beginning of an entry. The DS_ENTRYBEGIN directive is used in display or edit templates to mark the start of an LDAP entry and in list templates to mark the beginning of a section which should be repeated for each entry which is returned by the search. Always paired with DS_ENTRYBEGIN.

Arguments

None.

DS_ENTRYEND

Description

Delimits the end of an entry. Always paired with DS_ENTRYBEGIN.

Arguments

None.

DS_ATTRIBUTE

Description

The DS_ATTRIBUTE directive is replaced with the contents of an attribute (i.e., its values). This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

attr=attribute-name . Displays the named attribute. Any attribute may be displayed. The special attribute "dn" is recognized and causes the distinguished name of the entry to be displayed.

syntax=syntax-type . Displays the attribute as if it were of syntax syntax-type. If no syntax= argument is given, syntax=cis is assumed. Legal values are described in Table 7.5.

Table 7.5 DS_ATTRIBUTE: Display of Syntax Argument

Syntax Type	Description	Display as
tel	display as a telephone number	text
dn	display as a distinguished name	href (a link to an LDAP entry)
mail	display as a mailto: URL	href (mailto: URL)
mls	display as a multi-line string	text
time	display as date/time	text
cis	display as a case-ignore string	text
url	display as a labeled URL	href (URL)

type=how-to-display. Renders the attribute onscreen in a particular format. Legal values described in Table 7.6 correspond roughly to HTML form element names.

Table 7.6 DS_ATTRIBUTE: Display of Type Argument

Type Argument	Display as
text	display as text
textarea	show as an HTML TEXTAREA
radio	show as a radio button
checkbox	show as a check box
password	show as an HTML password text box (characters are not echoed)
hidden	show values in hidden form fields (Not supported in DS 1.0)

options=option. Modifies how the attribute is displayed. Legal values are described in Table 7.7.

Table 7.7 DS_ATTRIBUTE: Display of options Argument

Options	Display
sort	sorted attribute values
nolink	do not attempt to display the attribute as a hyperlink
dntags	only applies when using syntax=dn - tags are displayed when showing DNs. Normally they are not displayed
dateonly	only applicable when using syntax=time - only display the date, omitting the time
readonly	when editing, do not allow the user to modify the attribute's value
dnpicker	only applies when using syntax=dn - embed delete checkboxes and Javascript array info. needed for "Find and Add" (Not supported in DS 1.0)
unique	enforce uniqueness when adding or editing values (Not supported in DS 1.0)

defaultvalue=default-value . Supplies a default value for the attribute, which is shown if no attribute was read fromthe LDAP server.

within=string-to-embed-in. For each value, outputs the text in string-to-embed-in, replacing all occurrences ofthe string --value-- with an attribute value.

href=href. Specifies the HREF used for the hyperlink. For example, you can specify anonMouseOver JavaScript handler using the "href=" option.

hrefextra=extra-text. Specifies additional text which is inserted after the closing quote of the HREF tag.

dncomponents=number. Gives the number of DN components to show when displaying a DN. For example, if you include "dncomponents=2" and display the DN "cn=James Doe, o=Netscape Communciations Corp, c=US", the output will be "James Doe, Netscape Communications Corp.".

size=number. same as cols argument.

rows=number, rows=+number, rows=>number. Controls the number of rows used to display the entry. For type=text, this controls the number of editable HTML INPUT fields. For type=textarea, this controls the number of rows in the textarea. If number is preceded by a plus (+) sign, then number extra rows are included. If number is preceded by a greater-than sign, then at least number rows are included.

cols=number, cols=+number, cols=>number. Controls the width of the displayed attribute. If a number is given by itself, then the attribute is displayed with exactly number columns. If a plus (+) sign is given before number, then the attribute is given number extra columns. For example, if the value is 10 characters wide, and number is 10, then 20 columns are used when displaying the number. If a greater-than sign (>) is given before number, then the displayed width is at least number columns.

numfields=number, numfields=+number, numfields=>number. controls the number of editable fields displayed when editing. If number is preceded by a plus (+) sign, then the number of fields displayed is however many values were read from the server plus number. If number is preceded by a greater-than sign (>), then at least number values are displayed when editing.

true=string. label used for Boolean values that are true.

false=string. label used for Boolean values that are false.

value=string. value associated with an instance of a checkbox that is used to display strings values (not syntax=bool values) (Not supported in DS 1.0)

Examples

```
<!-- DS_ATTRIBUTE "attr=dn" "syntax=dn" "dncomponents=2"
"options=nolink" -->

<!-- DS_ATTRIBUTE "attr=givenName" "cols=>32" -->

<!-- DS_ATTRIBUTE "attr=sn" "cols=>32" -->

<!-- DS_ATTRIBUTE "attr=uid" "numfields=1" "cols=>16" "options=unique" -
->

<!-- DS_ATTRIBUTE "attr=mail" "syntax=mail" "cols=>20" -->

<!-- DS_ATTRIBUTE "attr=telephoneNumber" "syntax=tel" "cols=>16"
"numfields=+1" -->

<!-- DS_ATTRIBUTE "attr=modifyTimestamp" "syntax=time" "defaultvalue=N/
A" "options=readonly" -->

<!-- DS_ATTRIBUTE "attr=modifiersName" "syntax=dn" "defaultvalue=N/A"
```

```
"options=readonly" -->

<!-- DS_ATTRIBUTE "attr=mailDeliveryOption" "type=CHECKBOX"
"value=mailbox" -->

<!-- DS_ATTRIBUTE "attr=mailDeliveryOption" "type=CHECKBOX"
"value=native" -->

<!-- DS_ATTRIBUTE "attr=mailForwardingAddress" "syntax=mail"
"type=textarea" "rows=2" "cols=30" -->
```

DS_OBJECTCLASS

Description

Describes the type of directory entries a given template should be used for.

Arguments

value=value1,value2,...valueN. specifies a list of objectclass values. In order for a template file to be used todisplay a given entry, all of the values given must be values in the entry's objectclass attribute.

Note: the gateway does not read the template files to determine which template to use. Instead, it reads the dsgw.conf file and scans the "template" lines in that file. You can generate a series of template lines, suitable for inclusion in the dsgw.conf file, by opening the URL http://host/ds/templateindex, where "host" is the name of the host running the directory server gateway. The templateindex CGI program will scan all the Gateway template files and read the DS_OBJECTCLASS attributes, and then will generate a series of "template" lines.

Example

```
<!-- DS_OBJECTCLASS "value=person,inetOrgPerson" -->
```

DS_VIEW_SWITCHER

Description

Display a widget that provides access to all views that are appropriate for this entry (Not supported in DS 1.0). Usually this directive will be used without any arguments at all, which causes a table that contains one cell for each available view to be displayed.

Arguments

prefix=text. HTML text to emit before view elements (optional)

suffix=text. HTML text to emit after view elements (optional)

curprefix=text. HTML text to emit before the link to the current (active) view element (optional)

cursuffix=text. HTML text to emit after the link to the current view element (optional)

altprefix=text. HTML text to emit before each link to an alternative view element (optional)

altsuffix=text. HTML text to emit after each link to an alternative view element (optional)

Example

```
<!-- DS_VIEW_SWITCHER -->
```

DS_SORTENTRIES

Description

Specifies that entries should be sorted; typically used within list templates. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block. Up to two DS_SORTENTRIES directives are honored (the attribute from the first one that appears is used as the primary sort key and the second one is used as a secondary sort key).

Arguments

attr=attrname. Sort the entries in ascending order by attrname.

Example

To sort a list of entries by common name:

```
<!-- DS_SORTENTRIES "attr=cn" -->
```

DS_SEARCHDESC

Description

Specifies that text describing the type of search done should be displayed. For example, "Found 14 entries where the phone number ends with '25'".

Arguments

None.

DS_POSTEDVALUE

Description

Echoes the contents of an arbitrary posted form variable within a VALUE= parameter.

Arguments

name=varname. The name of the form variable.

Example

If a variable called searchstring is posted and contains the text Mark Smith, the directive:

```
<!-- DS_POSTED_VALUE "name=searchstring" -->
```

will produce the following

```
HTML: VALUE="Mark Smith"
```

DS_EDITBUTTON

Description

Displays a button which, when clicked, brings up an editable view of an entry. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block. Typically used in display templates.

Arguments

label=text. Use "text" as the label on the button. If not provided, the text "Edit" is used.

Example

```
<!-- DS_EDITBUTTON "label=Edit Person" -->
```

DS_DELETEBUTTON

Description

Displays a button which, when clicked, allows deletion of an entry. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block. Typically used in edit templates.

Arguments

label=text. Use "text" as the label on the button. If not provided, the text "Delete" is used.

Example

```
<!-- DS_DELETEBUTTON "label=Remove Person" -->
```

DS_SAVEBUTTON

Description

Displays a button which, when clicked, saves changes to an entry. Typically used in edit templates. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

label=text. Use "text" as the label on the button. If not provided, the text "Save" is used.

checksubmit=javascript. only submit changes if javascript expression is true.

Examples

```
<!-- DS_SAVEBUTTON "label=Save Changes" -->
```

```
<!-- DS_SAVEBUTTON "checksubmit=formDataValid()" -->
```

DS_EDITASBUTTON

Description

Displays a button which, when clicked, allows editing of an entry using a non-default template. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

label=text. Use "text" as the label on the button. If not provided, the text "Edit As" is used.

template=template-name. use the template name template-name when editing.

Example

A button to bring up edit-passwd.html template:

```
<!-- DS_EDITASBUTTON "label=Change Password" "template=passwd" -->
```

DS_NEWPASSWORD

Description

Displays an HTML password INPUT field. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

None.

DS_CONFIRM_NEWPASSWORD

Description

Displays an HTML password INPUT field. The gateway compares the value supplied by the user in this field to the value in the DS_NEWPASSWORD field, and only saves the new password value if the two match. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

None.

DS_OLDPASSWORD

Description

Displays an HTML password field for the old password. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

None.

DS_HELPBUTTON

Description

Displays a help button.

Arguments

topic=topic_name. causes the Netscape Help System to open the given topic name.

Example

```
<!-- DS_HELPBUTTON "topic=MODIFYPASSWD" -->
```

DS_CLOSEBUTTON

Description

Displays a Close button, which causes the containing window to be closed.

Arguments

label=text. Use "text" as the label on the button. If not provided, the text "Close Window" is used.

Example

```
<!-- DS_CLOSEBUTTON "label=Cancel" -->
```

DS_BEGIN_ENTRYFORM

Description

Causes the gateway to emit an HTML FORM directive, and several hidden form elements which are required for proper operation of the gateway. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

None.

DS_END_ENTRYFORM

Description

Causes the gateway to emit a </FORM> tag. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

None.

DS_EMIT_BASE_HREF

Description

Emit a <BASE> tag that contains the base URL for the CGI that was executed. (Not supported in DS 1.0)

Arguments

None.

DS_DNEDITBUTTON

Description

Used to edit DN-valued attributes, such as group member.

Arguments

label= .

template= .

attr= .

desc= .

DS_BEGIN_DNSEARCHFORM

Description

Used to edit DN-valued attributes, such as group member.

Arguments

None.

DS_END_DNSEARCHFORM

Description

XXX (Not supported in DS 1.0)

Arguments

None.

DS_ATTRVAL_SET

Description

display an attribute based on an "attrvset" as defined in the dsgw.conf file.

Arguments

set=name. use information from attribute value set name

prefix=text. HTML text to emit before each attribute value element (optional)

suffix=text. HTML text to emit after each attribute value element (optional)

Plus any of the arguments supported by the DS_ATTRIBUTE directive

Example

```
<!-- DS_ATTRVAL_SET "set=CAL" "attr=nsLicensedFor" "type=checkbox"
"prefix=<TR><TD>" "suffix=</TD></TR>" -->
```

IF/ ELSE/ ELIF/ ENDIF

Description

Set of directives that can be used to conditionally include HTML text

Arguments for IF and ELIF

condition. boolean condition; if true, include following block of text

!condition. boolean condition; if false, include following block of text

Arguments for ELSE and ENDIF

None.

Table 7.8 Conditions supported for ELSE and ENDI

Condition	Arguments	Description
FoundEntries	none	Are there any entries being displayed?
Adding	none	Is the entry being edited a new entry?
Editing	none	Are we editing an entry?
Displaying	none	Are we just displaying an entry?
Bound	none	Is the user authenticated?
BoundAsThisEntry	none	Is the user authenticated as the entry we are displaying?
AttributeHasValues	attr mincount	Does the attribute attr have at least mincount values? (Not supported in DS 1.0)
AttributeHasThisValue	attr syntax value	Does the attribute attr with synatx syntax have value as one of its values? (Not supported in DS 1.0)
AdminServer *	none	Are we running under the Administration Server? (Not supported in DS 1.0)
DirectoryIsLocalDB *	none	Is the Directory Service in use the LDAP local database? (Not supported in DS 1.0)
PostedFormValue *	name value	Is a form variable called name present that has value as its value? (Not supported in DS 1.0)

Note that conditions marked with an astericks (*) are supported in all the directory gateway CGIs, not just dosearch and edit.

Examples

```
<!-- IF "!DirectoryIsLocalDB" -->
The entry was last modified by <!-- DS_ATTRIBUTE "attr=modifiersName"
"syntax=dn" "defaultvalue=N/A" "options=readonly" -->
```

```
<!-- ENDIF ---->

<!-- IF "AttributeHasThisValue" "objectclass" "cis" "mailRecipient" -->
// this entry is a mail recipient... do something special here
<!-- ENDIF ---->
```

Miscellaneous Directives

BODY

Description

Emit HTML <BODY> element that includes color information. (Not supported in DS 1.0).

Arguments

extrahtml.

Examples

```
<!-- BODY  -->
```

```
<!-- BODY "onLoad=setDefaults()" -->
```

COLORS

Description

Set color information to be used in subsequent BODY directives. (Not supported in DS 1.0).

Arguments

html-color-info.

Example

```
<!-- COLORS "TEXT=#000000 BGCOLOR=#FFFFFF LINK=#FF0000 VLINK=#8000FF
ALINK=#FF0000" -->
```

TITLE

Description

Emit HTML <HEAD>, <TITLE>, and <BODY> elements. Supported by all directory gateway CGIs.

Arguments

title-string.

Example

```
<!-- TITLE "Search Results" -->
```

ENDHTML

Description

Emit </BODY></HTML> sequence

Arguments

None.

HELPBUTTON

Description

Display a Help button (same effect as DS_HELPBUTTON directive, but can be used from any gateway CGI) (Not supported in DS 1.0)

Arguments

topic.

Example

```
<!-- HELPBUTTON "MODIFYPASSWD" -->
```

INCLUDE

Description

Include the contents of another HTML file. Note that you cannot nest include directives. (Not supported in DS 1.0)

Arguments

filename. the name of the file to include. This is relative to the html/ directory where files such as display-inetorgperson.html are located.

Example

```
<!-- INCLUDE extra.html -->
```

INCLUDECONFIG

Description

include the contents of an HTML-based configuration file. Note that you cannot nest include directives.(Not supported in DS 1.0)

Arguments

filename. the name of the file to include. This is relative to the config/ directory where files such as dsgw.conf are located.

Example

```
<!-- INCLUDE dsgw-orgperson.conf -->
```

DS_LAST_OP_INFO

Description

Display a string that shows the result of the last domodify run. Note that this directive works onlys when the genscreen or edit CGIs are invoked via domodify's completion_javascript feature.

Arguments

prefix=prefix-text. text displayed before the last operation info.

suffix=suffix-text. text displayed after the last operation info.

Example

```
<!-- DS_LAST_OP_INFO "prefix=<P><FONT SIZE=%2B1>The user " "suffix=</
FONT>" -->
```

DS_LOCATIONPOPUP

Description

Emit an HTML form element that contains a list of all the o's and ou's that are in the directory. If there is only one, a hidden field is produced; otherwise an HTML select field is produced. (Not supported in DS 1.0)

Arguments

name=varname. the name of the form element that is emitted.

prefix=select_prefix. text that is output before a select element.

suffix=select_prefix. text output after a select element.

Example

```
<!-- DS_LOCATIONPOPUP "name=base" "prefix=Choose a searchbase" -->
```

DS_GATEWAY_VERSION

Description

Emit a string containing the version of the directory gateway CGI being executed. (Not supported in DS 1.0).

Arguments

None.

Example

```
<!-- DS_GATEWAY_VERSION -->
```

IF/ ELSE/ ELIF/ ENDIF

Same as those supported by the dosearch and edit CGIs, except that only conditionals marked with an asterix (*) are supported.

Appendix

C

CGI Usage

This appendix provides regular expression-type descriptions of the ways to invoke the CGIs in script files. Each regular expression is followed by the variables the expression can take on a POST. Uppercase words are variables. Lowercase words are literals.

auth

auth[?context=CONTEXT[&dn=DN]]

```
authasrootdn = "true"
```

or

```
escapedbinddn = DN
```

csearch

csearch[?context=CONTEXT[&file=FILE]] where FILE = type|attr|match|string|base

```
searchType = SEARCHTYPE
```

```
searchAttr = SEARCHATTR
```

dnedit

dnedit?CONTEXT=context&TEMPLATE=tmplname&DN=dn&ATTR=attrname&DESC=description

doauth

```
escapedbinddn = DN
```

```
authdesturl = AUTHDESTURL
```

```
password = PASSWORD
```

domodify

```
changetype = CHANGETYPE
```

```
dn = DN
```

```
newrdn = RDN
```

```
completion_javascript = COMPL_JS
```

```
newpasswd = NEW_PASSWD
```

```
passwd = PASSWD
```

```
newpasswdconfirm = NPCONFIRM
```

dosearch

dosearch?context=BLAH[&hp=host[:port]][&dn=baseDN][&ldq=LDAPquery]]

```
mode = MODE
```

```
searchstring = SEARCH
```

```
STRING
```

```
type = TYPE
```

```
base = BASE
```

```
attr = ATTR
```

```
match = MATCH
```

```
attr = ATTR
```

```
filterpattern = FILTERPATTERN
```

```
filterprefix = FILTERPREFIX
```

```
filtersuffix = FILTERSUFFIX

scope = SCOPE

searchdesc = SEARCHDESC
```

edit

edit?template&context=CONTEXT[&info=INFOSTRING][&ADD][&dn=DN][&dnattr=ATTRNAME&dndesc=DESCRIPTION]

(GETs only. No POSTs.)

lang

lang?context=CONTEXT[&file=FILE]

newentry

newentry[?context=CONTEXT[&file=FILE]]

```
entrytype = ET

entryname = EN

rdntag = RDNTAG

selectLocation = SL

dnsuffix = DNSUFFIX
```

search

search?context=CONTEXT[&file=string]

(GETs only. No POSTs.)

unauth

unauth[?conetxt=CONTEXT]

Appendix

D

Gateway User Help

This appendix provides instructions for using the Directory Server 4.0 gateway interface. The information supplied here is intended to be repurposed as help topics that can be made available to gateway users from the Help button on gateway forms. Topics include:

- Directory Tree Structure
- Searching the Directory
- Authentication
- Adding Entries Using the New Entry Form

Directory Tree Structure

The hierarchy of data in the directory can be represented by a tree. At the top of the directory tree is the root entry (or suffix). The root entry usually represents the organization entry for the directory (for example, o=airius.com). The directory may contain more than one root entry. Before adding new entries, make sure that you know which suffix the gateway is supporting.

Below the root entry are branches of the directory representing organizational units. For example:

```
o=Airius.com
```

```
ou=Marketing, o=Airius.com
```

```
ou=Accounting, o=Airius.com
```

Entries for people and resources within the organization appear below the organizational branches.

Distinguished Name Syntax

A directory entry is uniquely identified by its distinguished name (DN). The DN for an entry is represented as a series of comma-separated attributes and attribute values. The left-most value represents the entry's name. Each subsequent attribute represents a branch point above the entry. For example, this DN represents the entry named *malonso* in the subdirectory named *people* in the directory named *airius.com*:

```
uid=malonso, ou=people, o=airius.com
```

Unique Distinguished Names

The directory server interface will not permit duplicate entries. To avoid duplications, use distinguished names that be in with the person's user ID (uid) rather than the person's common name. Since email IDs are by definition unique within an organization, one common method is to use a person's email address as their user ID. For example:

```
jwatson@airius.com
```

becomes the DN

```
uid=jwatson, o=airius.com
```

Searching the Directory

The directory server interface provides two types of searches:

- Standard search selects an appropriate method of searching based on the value specified. For example, entering the search string "moz@airius.com" initiates a search for matching email addresses.
- Advanced Search provides a means for searching against specific entry attributes. For example, it is possible to search for users whose last names start with "K" and whose phone numbers end with "2110".

Both types of searches permit searching for any of the entry types described in Table 7.9.

Table 7.9 Entry Types That Can Be Specified

Type of Entry	Description
People	Entries that describe a person
NT People	Entries that describes an NT user.
Groups	Entries that describe a group. Groups may be defined System Administrators, Tech Writers, or all the people interested in Fishing, or all the Color Printers at the site. Groups can also contain other groups.
NT Groups	Entries that describe a group of NT users.
Organization	Entries that describe an organization. An organization is usually a single, large entity such as a corporation or a university. An organization represents a major, static, subdivision or branch of the directory.
Org-Units	Entries that describe an organization unit. Organizational units describe units within an organization, such as Accounting, Marketing, or Biology department.
Anything	Entries that match the specified search criteria. Anything useful when the entry you are searching for is not a person, group, or organization

After the directory server completes the search, the directory server interface displays the search results, which provide links to all matching entries. Clicking an entry displayed on the search results list displays detailed information about the entry. If the entry is a person, you can also choose to view the person's digital business card (vCard).

Performing a Standard Search

Depending on what is entered, Standard Search determines whether to find entries that exactly match the criteria, entries that contain the criteria, or entries that contain words or syllables that sound like the criteria. An LDAP search filter can also be used in the standard search field.

To perform a standard search:

1. Select the Standard Search tab. The Standard Search form appears.
2. Enter the value to find in the Search for field. This field is not case sensitive. Any of the following values can be entered:

- A name or part of a name
- a person's initials
- some or all of a phone number
- some or all of an email address
- an LDAP search filter

3. Click Submit. The form data is submitted to the directory server and the directory server searches for any entries that match exactly, match partially, or sound like the value supplied. Resulting matches are displayed in the search results table.

Name Search Functionality

When the search string specifies characters other than numbers or does not contain an at (@) symbol, the standard search attempts to find full names, first names, or last names that match exactly, match partially, or sound like the supplied value.

For example, specifying the string "son" could return:

- Gary Stevenson
- Sun Ra
- Sonny Jergensen

Name and Initial Search Functionality

When the search string specifies the following items in the following order, the standard search executes a search for a first initial followed by a last name:

- a single letter

- a character space, a period, or a space and a period together
- one or more characters

For example, specifying the string “P.Ande” could return

- Paul Andes
- Pam Anderson
- Paolo Andellini

Similarly, if the search string specifies the following items in the following order, the standard search executes a search for a first name followed by a last initial:

- more than one character
- a character space, a period, or a space and a period together
- a single character

For example, specifying the string “M.Pai” could return

- Michelle Painter
- Mai Pai
- Marty Pain

Approximate (or “sounds-like”) and substring searches are not performed in this case.

Searching for Phone Numbers

Standard search automatically searches for a phone number if the value entered consists only of numeric characters. This type of search is an “ends with” search. That is, the directory server searches for any phone numbers that end with the specified string. For example, entering the sting “123” results in the display of all phone number that end with 123. A single hyphen is permitted in the search string if at least one digit precedes it.

Searching for Email Addresses

Standard search searches for matching email addresses if an at (@) symbol is provided. Standard Search first searches for any email addresses that exactly match the value entered. For example, specifying the string “rafi@” could return the exact match:

- rafi@

or, if no match exists for “rafi@” in the directory, the search could return:

- rafi@aardvark.com
- rafi@acme.org

Using LDAP Search Filters

An LDAP search filter can be used to search for entries with a specific attribute value. The standard search assumes that any string containing an equal sign is an LDAP search filter. For example, specifying “cn=*eve*” will initiate a substring search for any common name containing the string “eve”.

When specifying attributes within an LDAP search filter, use the attribute label used by the directory server internally rather than the attribute field name as displayed in the gateway’s search results.

Advanced Search

The advanced search is designed to search for entries that have specific values for certain attributes (for example, a person entry whose email address is a specified value). Advanced search also allows searches for entries that do not include a specified attribute value (For example, all the people whose last name is not “Smith”).

Advanced search performs an exact search, returning entries that exactly match the words entered. There are four fields used to construct an advanced search. Constructed as follows, these four fields represent a sentence specifying the search:

Find: *[a type of entry]* where the: *[attribute]* [type of search] *[search string]*

The options for the first three of these fields are provided in pull-down menus. The fourth field contains the actual search string. For example:

Find: *[People]* where the: *[Last Name]* [is] *[Supriya]*

Find: *[People]* where the: *[Full Name]* [sounds like] *[Lloyd Daniels]*

Performing an Advanced Search

1. Select the Advanced Search tab. the The Advanced Search form appears.
2. Select the type of entry to search for.
3. Select the attribute to search for from the “where the” field pull-down menu. The choices available depend on the entry type selected in the Find field (see Table 7.10).
4. Select the matching type to use in the search.
5. Enter the search string in the text box and click “Search.” The form data is submitted and the directory server searches for entries that exactly match the value supplied. The resulting matches are displayed as a search results list.

Table 7.10 Advanced Search Attributes

Find Field	Searches Attributes That Can Be Specified
People or NT People	full name, last name, phone number, email address, user ID, or title
Groups or NT Groups	description, owner, or NT Group Type (for NT Groups)
Organizations	location, phone number, or description
Anything	name or description

Viewing Search Results

When search form data is submitted, the search results displayed depend on whether there were no matches, a single match, or multiple matches.

No matches

A search result that returns no matches means one of the following has occurred:

- No entries in the directory match the search criteria.
- You did not authenticate before performing the search.
- The access control for the directory subtree does not permit viewing of matching entries, regardless of authentication privileges.

A Single match

When a single match is returned in a response to an "is" search, the gateway displays all details for the entry.

Multiple matches

When multiple matches are found in response to a search, the gateway displays a table listing each of the matching entries and relevant information for each entry, such as the entry's phone number and email address.

Search Tips

Numeric Values

When search for a numeric value, such as a room number or a telephone number, be sure to include all spaces and leading zeroes.

Blank Spaces

The directory server interface strips all leading and trailing blank spaces from the search criteria.

Authentication

Authentication is the process of enabling users to perform operations on the directory. By default, access to the directory is denied to all users with the exception of the directory administrator. The directory administrator defines the user permissions that grant or restrict access to information in the directory.

Access Control

Using the access control mechanism, the directory administrator can allow or deny access:

- to any unauthenticated user (anonymous access)
- to all authenticated users
- to specific authenticated users or groups
- from a specific machine on DNS domain
- at a specific time of day or day of the week
- based on authentication method

Following are some of the access restrictions and access grants the directory administrator can set up by applying permissions to the directory:

- require the user to authenticate before accessing the directory in any way.
- require the user to authenticate before accessing certain subsections of the directory.
- require the user to authenticate before performing certain kids of actions, such as adding or modifying entries.
- deny user access to all or parts of the directory, or deny the ability to perform certain kinds of functions.
- allow anonymous access to all or parts of the directory.
- allow anonymous access for some kinds of operations (such as searches), but not others (such as modifications).

- allow or deny access based on the physical machine being used.

The interface prompts the user to authenticate before allowing modifications to the directory. A user who does not authenticate is allowed those permissions enabled for anonymous access.

For more information, see the *Netscape Directory Server Administrator's Guide* and the *Netscape Directory Server Deployment Guide*.

Authenticating to the Directory

Users can explicitly choose to authenticate by clicking the Authentication tab or wait until the DS interface automatically prompts for authentication before continuing with an operation. To authenticate to the directory:

1. Click on the Authentication tab. The Authentication tab appears.
2. Enter the name you want to use to identify yourself to the directory server. To authenticate as a regular user, enter your name as it appears in the directory server (common name or full name). Do not enter the user ID or login for a local operating system. To authenticate as the privileged directory user, click the Authenticate as Directory Manager button. The directory server displays a table of matching entries.
3. Select the link that corresponds to your directory entry (if the name is unique in the directory, the system skips this step). The system prompts for a password.
4. Enter the password and click Continue.
5. Click Return to main to continue to the default gateway.

Maintaining Authentication Credentials

By default, authentication credentials are set to last for 120 minutes before expiring. The expiration time is configurable by the directory administrator. When authentication credentials expire while a directory operation is being performed, re-authentication is necessary before completing the operation.

Logging Out of the Directory

To unauthenticate:

1. Click on the authentication tab. The Authentication form appears.
2. Click the Discard Authentication Credentials button. The user is returned to anonymous access privileges.

Troubleshooting Authentication Problems

The following table lists common authentication problems, possible causes, and actions that may be taken to resolve the problem.

Table 7.11 Authentication Problems

Problem	Possible Cause	Possible Action
Search results are empty.	No entries match the search string entered, or user authentication required.	Try a different search operation or authenticate to the directory.
Search results in missing entries or attribute information.	You are not authenticated properly or do not have privileges required to access the information (in which case the directory server responds as if the information does not exist).	Make sure you are properly authenticated. Verify with the system administrator that you have access to the directory information you need.
Operation fails after completion.	The directory is failing the operation because of improper authentication or because authentication has expired. This occurs because the LDAP protocol does not allow the interface to know whether authentication is required before trying an operation.	Make sure you are properly authenticated and that your authentication has not timed out.
A table of entries is displayed instead of a specific entry.	The full name is not unique or the name entered does not exist in the directory.	If your entry is not displayed, click Cancel and try to authenticate again. Be sure to use full name and not user ID.
User name is correct, but authentication fails anyway.	Password is incorrect.	Click retry and re-enter your password.

Adding Entries Using the New Entry Form

The New Entry form on the default gateway (dsgw.conf) can be used to add the following types of entries:

- a person

- an NT person
- a group
- an NT group
- an organizational unit
- an organization

Directory authentication is required before entries can be added to the directory using the gateway's New Entry form.

Adding a Person Entry

1. Click on the New Entry tab to bring up the New Entry form.
2. For type of entry, select Person.
3. Enter a user name for the person.
4. Specify a directory location for the entry. ou=People is the most common location for a new user. ou=Special Users can be chosen as a location for entries with more privileges than People entries. Choose Other to specify a DN for a directory location other than ou=People or ou=Special Users.
5. Click Continue to submit the new person entry (to cancel the operation, use the browser's Back button). The New Person window appears.
6. Enter values for all required fields. Full name and Last Name are required fields on the default gateway. (Values for Manager, Admin, and See Also can be added after the entry is saved.)
7. Click the Save New Person entry button at the top of the window. The gateway confirms that the new entry has been added and displays all information fields that can be modified.

Adding an NT Person Entry

When creating an NT Person entry, make sure that the subtree where the entry is placed is the subtree used by the directory's NT Synchronization Service to synchronize entries. When an NT Person entry is placed in another location, it is not synchronized with the Windows network

Required fields for an NT Person entry include:

- Last name
- Full Name
- NT Domain Name
- NT User ID

Adding a Group Entry

1. Click on the New Entry tab to bring up the New Entry form.
2. For type of entry, select Group.
3. Enter a name for the group.
4. Specify a directory location for the entry. ou=Groups is the common location for group entries. ou=Special Users can be chosen as a location for entries with more privileges than People entries. Choose Other to specify the DN for a directory location other than ou=People or ou=Special Users.
5. Click Continue to submit the new group entry (to cancel the operation, use the browser's Back button). The New Group window appears.
6. Enter a value for Name. (Values for Owner, See Also, and Group Members can be modified after the entry has been saved.)
7. Click the Save New Group button at the top of the window. The gateway confirms that the new entry has been added and displays all information fields that can be modified.

Adding an NT Group

Required fields for an NT Group include:

- Name
- NT Group Name
- NT Group Domain

Adding an Organizational Unit Entry

1. Click on the New Entry tab to bring up the New Entry form.
2. For type of entry, select Organizational Unit.
3. Enter a name for the organizational unit.
4. Specify a directory location for the entry. ou=People is a common location for adding new organizational units. Choose This Organization to specify the DN for an organizational unit directly under the root entry. Choose Other to specify the DN for a location under a different root entry.
5. Click Continue to submit the new organizational unit entry (to cancel the operation, use the browser's Back button). The New Organizational Unit window appears.
6. Fill in the information fields for the new organizational unit and click the Save New Organizational Unit button at the top of the window. The gateway confirms that the new entry has been added and displays all information fields that can be modified.

There are many methods of setting up the directory tree structure for an enterprise. For detailed information, refer to the *Netscape Directory Server Deployment Guide*.

Adding an Organization Entry

An organization can only be added when the directory is initially populated. Organizations added must match the directory tree structure specified during directory server installation. The New Entry form can not be used to create a new root entry.

1. Click the New Entry tab to bring up the New Entry form.

2. For type of entry, select Organization.

3. Enter a name for the organization.

4. Specify a directory location for the entry. The DN specified must appear under the root entry specified during directory server installation. For example, if the root entry is o=Airius.com, then the DN must include o=Airius.com.

5. Click Continue to submit the new organization entry (to cancel the operation, use the browser's Back button). A New Organization window appears.

6. Fill in the information fields for the new organization and click the Save New Organization button at the top of the window. The gateway confirms that the new entry has been added and displays all information fields that can be modified.

Index