

Gateway Customization Guide

iPlanet Directory Server

Version 5.0

8xx-xxxx-xx
April 2001

Copyright © 2001 Sun Microsystems, Inc. Some preexisting portions Copyright © 2001 Netscape Communications Corporation. All rights reserved.

Sun, Sun Microsystems, the Sun logo, iPlanet, and the iPlanet logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Netscape and the Netscape N logo are registered trademarks of Netscape Communications Corporation in the U.S. and other countries. Other Netscape logos, product names, and service names are also trademarks of Netscape Communications Corporation, which may be registered in other countries.

Portions of the Software copyright © 1995 PEER Networks, Inc. All rights reserved. The Software contains the Taligent International Classes from Taligent, Inc. and IBM Corp. Portions of the Software copyright ©1992-1998 Regents of the University of Michigan. All rights reserved. The software contains encryption software from RSA Security Inc. Copyright © 1994 RSA Data Security, Inc. All rights reserved.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of the Sun-Netscape Alliance and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2001 Sun Microsystems, Inc. Pour certaines parties préexistantes, Copyright © 2001 Netscape Communications Corp. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, iPlanet et le logo iPlanet sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d'autre pays. Netscape et the Netscape N logo sont des marques déposées de Netscape Communications Corporation aux Etats-Unis et d'autre pays. Les autres logos, les noms de produit, et les noms de service de Netscape sont des marques déposées de Netscape Communications Corporation dans certains autres pays.

Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite préalable de l'Alliance Sun-Netscape et, le cas échéant, de ses bailleurs de licence.

CETTE DOCUMENTATION EST FOURNIE “EN L'ÉTAT”, ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

Preface	11
Audience	11
Typographic Conventions	11
Related Information	12
Chapter 1 Introduction	1
What is a Gateway?	1
Gateways Installed with Directory Server 4.0	2
Default Gateway	2
Directory Express	2
HTTP Server Requirements for Gateways	2
Directory Express and Default Gateway	3
Directory Express (pb.conf)	3
Default Gateway (dsgw.conf)	4
Support for Multiple Gateway Instances	5
Specifying Gateway Configuration to Gateway CGIs	5
GET Operations (GCONTEXT)	5
POST Operations (PCONTEXT)	6
URLs for Gateway Access	6
Release 4.0	6
Release 3.x	6
Gateway Localization	7
Gateway Locales	7
Character Sets	7
Non-Anonymous Searching	7
binddnfile	8
Compatibility with Existing Gateways	8
Automatic Updates to Directory Configuration	8

Chapter 2 Setting Up the Gateway	11
Gateway Installation Planning	11
Location of Gateway Files	11
Gateway Release 4.0	11
Gateway Release 3.x	12
Securing 4.0 Gateway Configuration and Settings	13
Protecting Bind DN and Password	13
Protecting Root Processes on UNIX Systems	13
Migrating 3.x Default Gateways to Release 4.0	13
Location of migrated dsgw.conf file	14
Running the Gateway Migration Script	14
Updating the Gateway with Changes to Directory Server Configuration	14
HTTP Server Recommendations for Directory Server Gateway Release 4.0	15
Running the Gateway in High-Usage Networks	15
HTTP Server Configuration	16
Name Translation Mapping	16
Gateway Root Suffix	16
Configuring the Gateway for Enterprise or FastTrack Servers	17
Add an Additional Document Directory (4.0 Gateways)	18
Add an Additional Document Directory (3.x Gateways)	19
Add an Additional CGI Directory (for 4.0 Gateways)	19
Add an Additional CGI Directory (for 3.x Gateways)	19
Change Permissions of Cookie Directory (UNIX only)	20
Creating a New Gateway Instance (4.0 only)	20
Gateway Cloning	21
Gateway .conf File Configuration	21
Changing the Default Port Setting	22
Setting Up the Directory Manager	22
Configuring the Directory Manager DN	22
Setting up the Suffix for Adding Entries	23
Setting Up SSL Support	24
Configuring the Gateway to Use SSL	24
Setting Up Localization	24
Setting vCard Properties	25
Configuring Gateway Clients	25
Language Support for HTTP Clients	25
Unicode and Latin-1 Character Sets	25
Displaying a Non-English Alphabet	26
Configuring Communicator 4.x for Preferred Language	26
Configuring Navigator 3.01 for Preferred Language	26
Customizing Communicator's LDAP Settings	27

Chapter 3 Gateway Localization	29
Unicode and iPlanet Support for UTF-8	29
How the Gateway Selects a Character Set	30
How the Gateway Selects from Multiple Requested Characters Sets	30
HTTP Clients that Request UTF-8	30
HTTP Clients that Do Not Request UTF-8	31
Special Characters	31
Non-Breaking Space	31
Query Strings	31
Gateway Locales	32
Support for Multiple Locales	32
Setting Up Locales for Translation	33
dsgw-l10n.conf	33
Chapter 4 File Controlling Gateway Functionality	35
Files Controlling Gateway Functionality	35
Gateway .conf Files	36
dsgw.conf	37
pb.conf	37
Gateway Search Configuration Files	37
dsgwsearchprefs.conf	37
dsgwfilter.conf	37
Object Class Templates	38
Gateway Script Files	38
Gateway Search Result Templates	39
Banner Files	40
Chapter 5 Editing Entry Types	41
Entry Types (Object Classes)	41
Parameters Controlling Entry Types	41
template	42
location	42
newtype	42
Considerations for Adding New Entry Types	43
Mapping Locations and Entry Types	44
Mapping Entry Types to Locations	44
Setting Up Organizational Units	45
Configuring DN Formats for Entry Types	45
UID-Based DN	45
Modifying the Default DN format	45
Object Class Attributes in Template Files	46
Default Gateway Object Classes	46
Templates and Directives	46

Entry-Related Directives	46
Adding Attributes to Object Classes	47
Deleting Attributes from Object Classes	47
Extending Object Classes	48
Adding a Template for a Child of a Parent Class	49
Creating a New Parent Object Class	50
Chapter 6 Search Attributes, Filters, and Results	53
Search Configuration Files	53
dsgwsearchprefs.conf	53
dsgwfilter.conf	54
Changing Search Scope	54
Modifying Search Attributes for Advanced Searches	55
Standard and Advanced Searches	55
Specifying Search Attributes for Person	57
Directory Express Search Support for user ID	58
Adding Search Support for Additional Attributes	58
Adding Search Support for a New Object	59
Extending Search Preferences	59
Creating a New Search Object	60
Modifying Default Search Filters	61
Search Filters for User Data Patterns	62
Specifying a Search Filter for a New Object	62
Customizing Search Result Templates	63
How the Gateway Displays Search Results	63
Search Result Tables	63
Modifying Search Result Templates	65
Adding Information to Search Results	65
Removing Information From Search Results	66
Chapter 7 Customizing Graphics and Color	67
Appearance of Gateway Forms	67
Banner Image	68
Updating the Banner Image (title.gif)	68
Changing Dimensions of Banner Image	68
Changing Filename of Banner Image	68
Button Images	69
Updating Button Images	69
Color Schemes	70
Files Controlling Colors on Gateway Forms	70
Changing Colors Using BODY Tag	71
Changing Colors Using Directives	71

Changing Table Colors	72
Specifying Color Names and Color Codes	72
Changing Color of Table Headings	73
Changing the Background Color of Table Rows	73
Appendix A .conf Parameters	75
authlifetime	75
Format	75
Example	75
baseurl	76
Format	76
Example	76
binddnfile	77
Format	77
Example	77
changeHTML	77
Format	77
Example	77
charset	77
Format	78
Example	78
configdir	78
Format	78
Example	78
dirmgr	79
Format	79
Example	79
gwnametrans	79
Format	79
Example	79
htmldir	79
Format	80
Example	80
ignoreAccetpCharsetFrom	80
Format	80
Example	80
include	80
Format	80
Example	80
location	80
Format	81
Example	81
location-suffix	81

Format	82
Example	82
newtype	82
Format	82
Example	83
NLS	83
Format	83
Example	83
securitypath	83
Format	83
Example	84
template	84
Format	84
Example	84
vcard-property	84
Format	84
Example	86
Appendix B Gateway Directives	87
Introduction	87
Structure of an HTML Template	88
Structure of an HTML Template for Directory List	88
Context-Related Directives	89
GCONTEXT	89
PCONTEXT	89
Entry-related Directives	89
DS_ENTRYBEGIN	90
DS_ENTRYEND	90
DS_ATTRIBUTE	90
DS_OBJECTCLASS	93
DS_VIEW_SWITCHER	94
DS_SORTENTRIES	95
DS_SEARCHDESC	95
DS_POSTEDVALUE	95
DS_EDITBUTTON	96
DS_DELETEBUTTON	96
DS_SAVEBUTTON	96
DS_EDITASBUTTON	97
DS_NEWPASSWORD	97
DS_CONFIRM_NEWPASSWORD	97
DS_OLDPASSWORD	97
DS_HELPBUTTON	98
DS_CLOSEBUTTON	98

DS_END_ENTRYFORM	98
DS_EMIT_BASE_HREF	98
DS_DNEDITBUTTON	99
DS_BEGIN_DNSEARCHFORM	99
DS_ATTRVAL_SET	99
IF/ ELSE/ ELIF/ ENDIF	99
Miscellaneous Directives	101
BODY	101
COLORS	101
TITLE	101
ENDHTML	102
HELPBUTTON	102
INCLUDE	102
INCLUDECONFIG	102
DS_LAST_OP_INFO	103
DS_LOCATIONPOPUP	103
DS_GATEWAY_VERSION	103
IF/ ELSE/ ELIF/ ENDIF	104
Appendix C CGI Usage	105
auth	105
lang	105
search	105
csearch	106
unauth	106
dnedit	106
edit	106
doauth	106
domodify	106
newentry	107
dosearch	107

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.

Audience

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.

Typographic Conventions

The following table lists the typographic conventions used in this guide:

Table 1 Typographic Conventions in the Gateway Customization Guide

Convention	Example	Comment
Pathnames	<code>http://host:port/dsgw/bin/lang?context=CONTEXT</code>	Pathnames specified in this guide are displayed in UNIX format.
4.0 gateway installation directory	<code>/usr/iplanet/servers/dsgw</code>	Represents the installation directory for the Directory Server instance on the host machine.
3.x gateway installation directory	<code><NSHOME>/slapd-server/dsgw</code>	Represents the release 3.x gateway installation directory for the Directory Server instance on the host machine.
Configuration file	<code>pb.conf</code> <code>dsgw.conf</code>	The .conf file represents the primary configuration file for a gateway instance.

Table 1 Typographic Conventions in the Gateway Customization Guide

Convention	Example	Comment
File syntax	People “ “Search for”: “(&(objectClass=person) ” “dsgw-people”	Gateway HTML file syntax is displayed on a distinct line in a monospaced font.
Command line syntax	# chown <uid> authck	Command line syntax to be entered by the user is displayed on a separate line in a monospaced font.

Related Information

The document set for iPlanet Directory Server also contains the following guides:

iPlanet Directory Server Installation Guide. Procedures for installing your Directory Server as well as procedures for migrating your Netscape Directory Server to iPlanet Directory Server.

iPlanet Directory Server Deployment Guide. Provides an overview for planning your deployment of the Gateway Customization Guide. Includes deployment examples.

iPlanet Directory Server Command and File Reference. Provides reference information on the command-line scripts, configuration attributes, and log files shipped with Directory Server.

iPlanet Schema Reference. Information about all the schema used in the iPlanet suite of products.

Other useful iPlanet information can be found at the following Internet locations:

- iPlanet release notes and other documentation:
<http://docs.iplanet.com/docs/manuals/>
- iPlanet product status:
http://www.iplanet.com/support/technical_resources/
- iPlanet Professional Services information:
http://www.iplanet.com/services/pro_serv/index.html
- iPlanet developer information:
<http://developer.iplanet.com/>

- iPlanet learning solutions:
<http://www.iplanet.com/learning/index.html>
- iPlanet product data sheets:
<http://www.iplanet.com/products/index.html>

Related Information

Introduction

This chapter describes the gateway functionality introduced with release 4.0 of the directory server. Topics include:

- What is a Gateway?
- Directory Express and Default Gateway
- Support for Multiple Gateway Instances
- Gateway Localization
- 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. Using a gateway does not require login to the iPlanet Console.

In release 4.0 of 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 `/usr/iplanet/servers/dsgw/context`, defining the context for 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.

Gateways Installed with Directory Server 4.0

Two gateway instances are installed during directory server installation: the default gateway and Directory Express. Both gateways are configured to use the suffix set when the directory server was configured, and non-SSL (Secure Socket Layer) communications.

For more information, see “Location of Gateway Files” on page 11.

Default Gateway

The configuration file for the default gateway is at
`/usr/iplanet/servers/dsgw/context/dsgw.conf`.

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
```

Directory Express

The configuration file for Directory Express is
`/usr/iplanet/servers/dsgw/context/pb.conf`.

During directory server installation, Directory Express is configured to use as its HTTP server the iPlanet 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
```

HTTP Server Requirements for Gateways

A gateway instance requires an HTTP server that can communicate 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 iPlanet Web Server, Enterprise or FastTrack edition.

For more information, see “HTTP Server Configuration” on page 16.

Directory Express and Default Gateway

The following sections describe directory express and the default gateway in more detail.

Directory Express (pb.conf)

Directory Express is a basic directory lookup tool that can be used out of the box.

Table 1-1 Directory Express Search Result








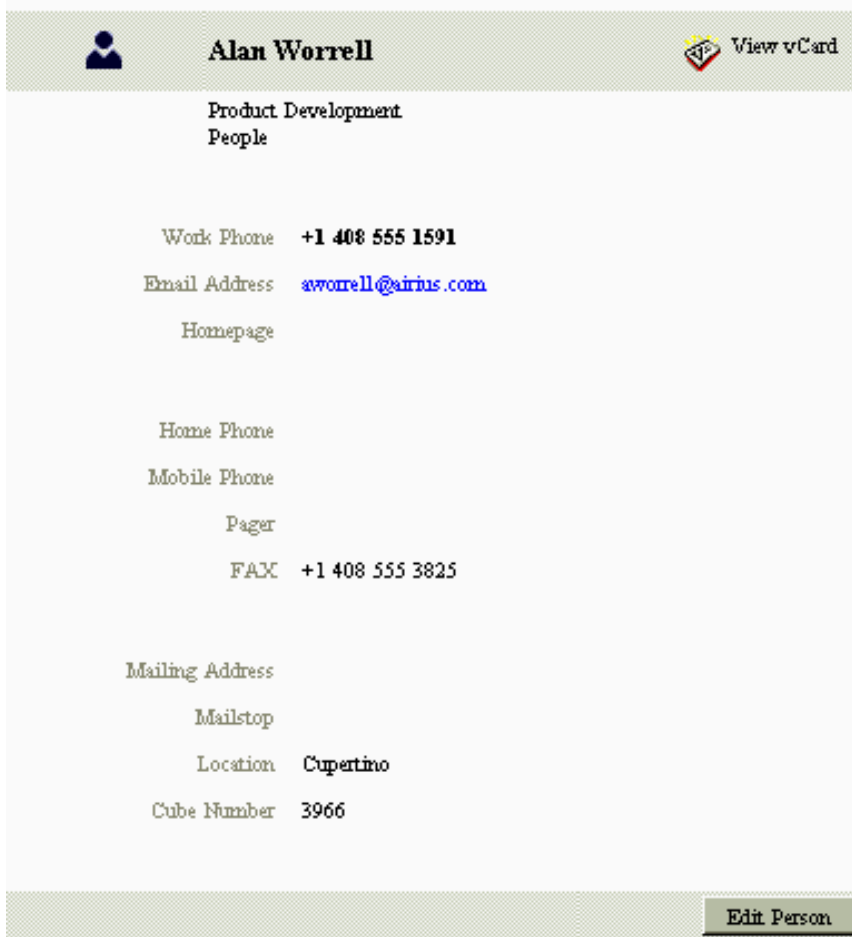


Netscape Directory Express				
<input type="text" value="castle"/>		<input type="button" value="Search"/>		
Enter any part of a name, user ID, or phone number in the search field.				
Found x entries where the name or user ID matches ' castle '.				
Name	ID	Phone	E-mail	Group
 Jane Castle	jane	4444	jane@arius.com	engineering
 John Castle	john	5555	john@arius.com	marketing
 King's Castle	King's Castle	2121		
 Quincy Castlegate	quincy	7777	quincy@arius.com	product development
 John Castle	john	5555	john@arius.com	marketing
 King's Castle	King's Castle	2121		
 Quincy Castlegate	quincy	7777	quincy@arius.com	product development

Table 1-2 Directory Express Extended Search Results

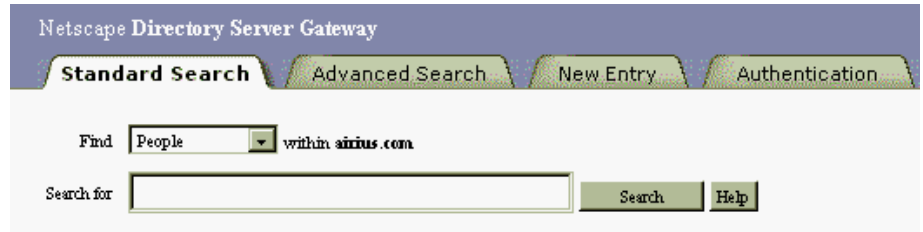


The screenshot displays a user profile for Alan Worrell. At the top left is a person icon. The name "Alan Worrell" is prominently displayed in the center. To the right is a "View vCard" link with a vCard icon. Below the name, the text "Product Development" and "People" is shown. The profile lists several contact details: "Work Phone" (+1 408 555 1591), "Email Address" (aworrell@airius.com), "Homepage", "Home Phone", "Mobile Phone", "Pager", and "FAX" (+1 408 555 3825). Under the "Mailing Address" section, it lists "Mailstop", "Location" (Cupertino), and "Cube Number" (3966). At the bottom right, there is an "Edit Person" button.

	Alan Worrell	 View vCard
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	
Edit Person		

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.

Table 1-3 Default Gateway

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 /usr/iplanet/servers/dsgw/context. Within the .conf file are two parameters specifying the path names for the HTML and template files for the gateway. The following lines show the HTML and configuration directories specified in the pb.conf file:

```
htmlmdir ../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 following example. 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. An example of embedding the GCONTEXT string in a link follows:

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

NOTE In Directory Server Gateway 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. For example, you can specify `PCONTEXT` in an HTML form as follows:

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

For more information about content-related directives, see “Context-Related Directives” on page 89.

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 iPlanet 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. The following sections describe the different locales and characters sets supported by the gateway.

Gateway Locales

Release 4.0 of the gateway is localized for English, Japanese, French, Spanish, and German. You can configure the gateway to support additional locales.

Language files are stored in `/usr/iplanet/servers/dsgw/html/lang` and `/usr/iplanet/servers/dsgw/config/lang`, where *lang* is defined in RFC1766.

For example, language files for Japanese are stored in `/usr/iplanet/servers/dsgw/html/ja` and `/usr/iplanet/servers/dsgw/config/ja[true]`.

Support for the character sets necessary to render a particular locale (language) must be available in the browser's configuration.

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.

For more information about gateway character sets, see "How the Gateway Selects a Character Set" on page 30.

Non-Anonymous Searching

Release 3.0 of the gateway allows anonymous searching only. This provides 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 `binddnfile`, can be set up for users to authenticate to the directory server. User permissions for directory access can be defined in the iPlanet Console.

If the directory server contains authentication credentials for a user, 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 `binndnfile`. When no `binndnfile` is specified, the gateway instance binds anonymously.

binndnfile

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

The `binndnfile` contains highly sensitive information. Do not store the `binndnfile` under `/usr/iplanet/servers/dsgw` or in any directory that is served up over HTTP (for instance, `/bin/slapd/server` is a good place to store the `binndnfile`).

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 13.

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 `/usr/iplanet/servers/dsgw/context` for gateway instances that match the directory server host and port.

The `updatedsgw` script runs automatically for gateways installed on the iPlanet 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 iPlanet Administration server managing the directory server instance runs the `updatedsgw` script.

For more information, see “Updating the Gateway with Changes to Directory Server Configuration” on page 14.

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

The following sections describe the steps for planning your installation of the gateway.

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	Gateway release 4.0
default gateway .conf file	/usr/iplanet/servers/dsgw/context/dsgw.conf
default gateway (dsgw) HTML and template files	/usr/iplanet/servers/dsgw/html /usr/iplanet/servers/dsgw/config
Directory Express .conf file	/usr/iplanet/servers/dsgw/context/pb.conf
Directory Express (pb) HTML and template files	/usr/iplanet/servers/dsgw/pbhtml /usr/iplanet/servers/dsgw/pbconfig

Gateway Cloning

Unique gateway instances may have unique HTML directories (for example, ..dsgw/myhtml) 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.

For more information on cloning the gateway, see “Gateway Cloning” on page 21.

Gateway Release 3.x

One gateway is installed for each instantiation of the directory server. The .conf file is stored in /dsgw (under *slapd-serverID*).

Table 2-2 Location of gateway files for release 3.x

File types	Gateway release 3.x
default gateway .conf file	/usr/iplanet/servers/ <i>slapd-serverID</i> /dsgw/dsgw.conf
default gateway HTML and template files	/usr/iplanet/servers/ <i>slapd-serverID</i> /html /usr/iplanet/servers/ <i>slapd-serverID</i> /config

Securing 4.0 Gateway Configuration and Settings

The following sections describe procedure for protecting the configuration information of your gateway.

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 (`/usr/iplanet/servers/dsgw`), or in any directory that is served up over HTTP.

Protecting Root Processes on UNIX Systems

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

Migrating 3.x Default Gateways to Release 4.0

The 3.x gateway will not be able to 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 it's 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 `/usr/iplanet/servers/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 `htmlmdir`, `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 `/usr/iplanet/servers/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 `/usr/iplanet/servers/bin/slapd/admin/bin`.

Changes made to the directory server configuration (`slapd.conf`) by the iPlanet Console are posted to `updatedsgw` and the relevant gateway files are updated. These files will be updated only 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 iPlanet Administration Server is the default HTTP server for the two gateway clients that are installed with the iPlanet Directory Server. Both Directory Express and the default gateway are pre-configured to run under the iPlanet Administration Server without additional setup.

There are many factors affecting 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 iPlanet Administration Server will begin 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 iPlanet Administration server that is also running a iPlanet server process as root. This may expose sensitive information about the configuration of iPlanet servers.

Running the Gateway in High-Usage Networks

Network administrators expecting high gateway usage may wish 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, we recommend iPlanet Enterprise Server or FastTrack Server.

HTTP Server Configuration

The following sections describe the steps for configuring an HTTP server.

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).

For more information on configuring the gwnametrans parameter, see "gwnametrans" on page 79.

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=siroe.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 to propagate the change to all gateway instances.

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 by iPlanet Console).

Configuring the Gateway for Enterprise or FastTrack Servers

The configuration procedures provided in this section assume that a iPlanet Web Server, FastTrack or Enterprise edition, is installed and configured to communicate with the iPlanet 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 iPlanet 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 iPlanet FastTrack Server configuration screen used to configure an additional document directory. Figure 2-2 shows the iPlanet Enterprise Manager configuration screen used to configure an additional CGI directory.

Figure 2-1 Configuring an Additional Document Directory

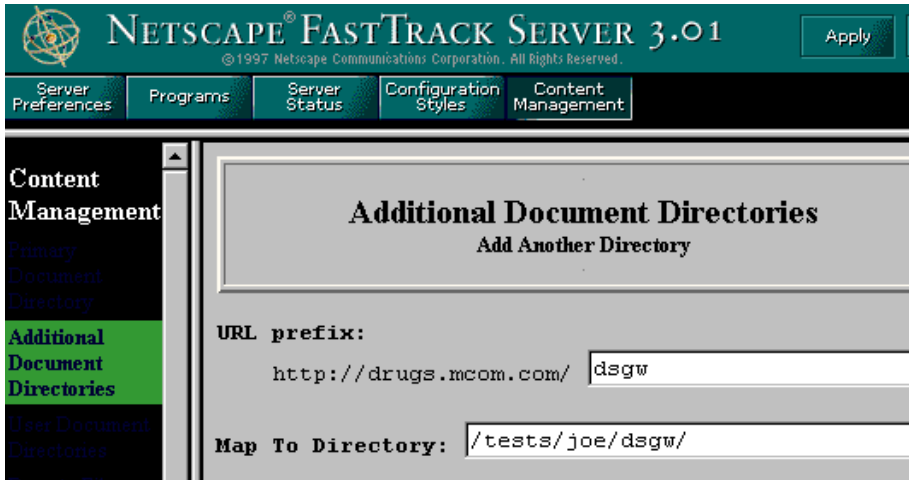
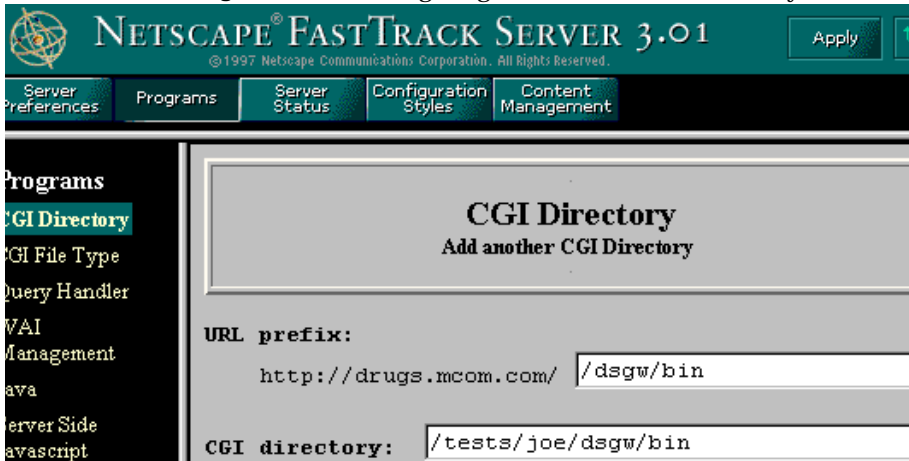


Figure 2-2 Configuring an Additional CGI Directory



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
`/usr/iplanet/servers/dsgw/`
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:
`/usr/iplanet/server/slappd-serverID/dsgw/html`
where *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:
`/usr/iplanet/servers/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:

```
/usr/iplanet/servers/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 `/usr/iplanet/servers/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 hostname, 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 will run under the iPlanet Administration server or a similarly capable HTTP server.

1. Rename `dsgw.conf` or `pb.conf` to a new gateway context. For example, `dsgw/context/dsgw.conf` might become `dsgw/context/siroe.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 `siroe.conf` should point to `/dsgw/sirohtml`.
3. To support non-anonymous searching (one individual user DN and password per directory instance) using the new gateway, set the `binddnfile` parameter in `siroe.conf` to point to the location of the file containing the bind DN and bind password that will 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 `siroe.conf`, copy and rename an existing HTML directory (`dsgw/html` or `dsgw/pbhtml`) to `/dsgw/sirohtml`.
5. Create a template directory containing object class templates and other configuration files. For example, to provide a template directory for `siroe.conf`, copy and rename an existing template directory (`/dsgw/config` or `/dsgw/pbconfig`) to `/dsgw/siroconfig`.
6. Edit the `htmlmdir` and `configdir` parameters in `siroe.conf` to point to the new HTML and template directories.
7. To access the new gateway instance--in this example, `siroe.conf`--navigate the browser to

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

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

The following sections describe the steps for configuring the gateway `.conf` file.

Changing the Default Port Setting

The LDAP port is set during directory server installation. This value can be changed in the `baseurl` parameter. The following example shows the syntax used to specify a port number that is different than the default port number of 389. For example, the `baseurl` parameter in the LDAP port is changed to the following:

```
baseurl "ldaps://dirserver.siroe.com:3000/o%3Dsiroe.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, set 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 will manage.
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 will manage.

The `dirmgr` parameter is described in “`dirmgr`” on page 79. Creating directory entries is described in the *iPlanet Directory Server Administrator's Guide*.

Authenticating as Directory Manager

Figure 2-3 shows 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 80.

Figure 2-3 Authenticating as Directory Manager

Netscape Directory Server Gateway

Standard Search Advanced Search New Entry **Authentication**

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 80. The *iPlanet Directory Server Administrator’s Guide* describes the Suffix parameter and provides syntax examples. Setting the root suffix is also described in the *iPlanet 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 hostname 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 *iPlanet Directory Server Administrator's Guide*. Information about managing key and certificate databases is provided in *Managing iPlanet Servers*.

Configuring the Gateway to Use SSL

The `securitypath` parameter specifies the location of the certificate database. For example, you can specify the path to the certificate database as follows:

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

The following example shows the `baseurl` parameter configured to use `ldaps` (instead of `ldap`, the default) and standard SSL port number 636:

```
baseurl "ldaps://dir.siroe.com:636/o%3Dsiroe.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.

For more information about the `baseurl` parameter, see “`baseurl`” on page 76.

Setting Up Localization

There are two considerations for configuring the gateway character set: what the directory contains, and the HTTP clients. The ideal character set supports all the characters in the directory, and will be 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 vCard Properties

Mappings between VCARD properties and LDAP attribute type are described in "vcard-property" on page 84.

Configuring Gateway Clients

The following sections describe how to configure clients of the gateway.

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 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 list, select a Unicode font set (for example, Bitstream Cyberbit).
5. From the Fixed Width Font pull-down list, 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 list, 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.

Gateway Localization

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

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

Unicode and iPlanet Support for UTF-8

Unicode is a character set containing all the characters of all the world's major languages. There are different standard methods to encode 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.

iPlanet products use UTF-8 in versions 2 and 3 of LDAP. Most software included in the directory server 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 charset 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 charset 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 Characters 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 pre-configured 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 pre-configuration to be overridden using the `ignoreAcceptCharsetFrom` parameter. For more information about this parameter, see "ignoreAccepCharsetFrom" on page 80.

The conversion from UTF-8 to the gateway client's chosen charset 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

The following sections describe how special characters are interpreted by the gateway.

Non-Breaking Space

If the client's charset lacks a character for non-breaking space, but has ideographic space, non-breaking spaces are converted to ideographic spaces before charset 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 charset 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
- Spanish
- 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, Spanish, and Japan. This is part of the overall localization effort, which includes localizing all the configuration and HTML files, including the on-line 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 `/usr/iplanet/servers/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-l10n.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.

The following example shows how to create a new locale using Chinese as the language for translation:

1. Create a “zh” directory in `/usr/iplanet/servers/dsgw/context`
2. Copy `dsgw.conf` to the `/usr/iplanet/servers/dsgw/context/zh`
3. uncomment this line from the gateway’s .conf file:


```
include "../config/dsgw-l10n.conf"
```
4. create a “zh” directory in `/usr/iplanet/servers/dsgw/config`
5. Copy or create the file `dsgw-l10n.conf`, stored during gateway installation in `/usr/iplanet/servers/dsgw/config/lang`, to `/usr/iplanet/servers/dsgw/config/zh`

NOTE If you are using the US version of the gateway, `dsgw.conf` contains a sample of `dsgw-l10n.conf`.

File Controlling Gateway Functionality

This chapter provides examples of customized gateways. Topics covered include:

- Files Controlling Gateway Functionality
- Gateway .conf Files
- Gateway Search Configuration Files
- Object Class Templates
- Gateway Script Files
- Gateway Search Result Templates
- Banner Files

Files Controlling Gateway Functionality

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

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

Gateway .conf Files

A gateway's .conf file describes an instance of the gateway. 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

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

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

pb.conf

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

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

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 the 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 dsgwsearchprefs.conf file specifies the object classes and object class attributes that can be searched.

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

dsgwfilter.conf

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

The dsgwfilter.conf file 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.

Modifying template files is discussed in Chapter 5, “Editing Entry Types.”

Default gateway object class templates stored in the `dsgw/conf` directory are listed in the following table.

Table 4-2 Default Template Files and Related Object Classes

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

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

Table 4-4 Default Search Result Templates

Search Result Template	Related Object Class
list-People.html	People

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

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

Editing Entry Types

This chapter describes how entry type formats--defined by object class and their attributes--can be controlled by editing 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)

The following sections describe entry types in more detail.

Parameters Controlling Entry Types

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

- Template parameter settings determine the types of objects that can be created and the attributes supported for object classes.
- Newtype parameter settings determine the DN formats to be applied to new entries.
- Location parameter settings determine where in the directory new entries reside.

Parameters in the `dsgw.conf` file are described in Appendix A, “.conf Parameters.”

template

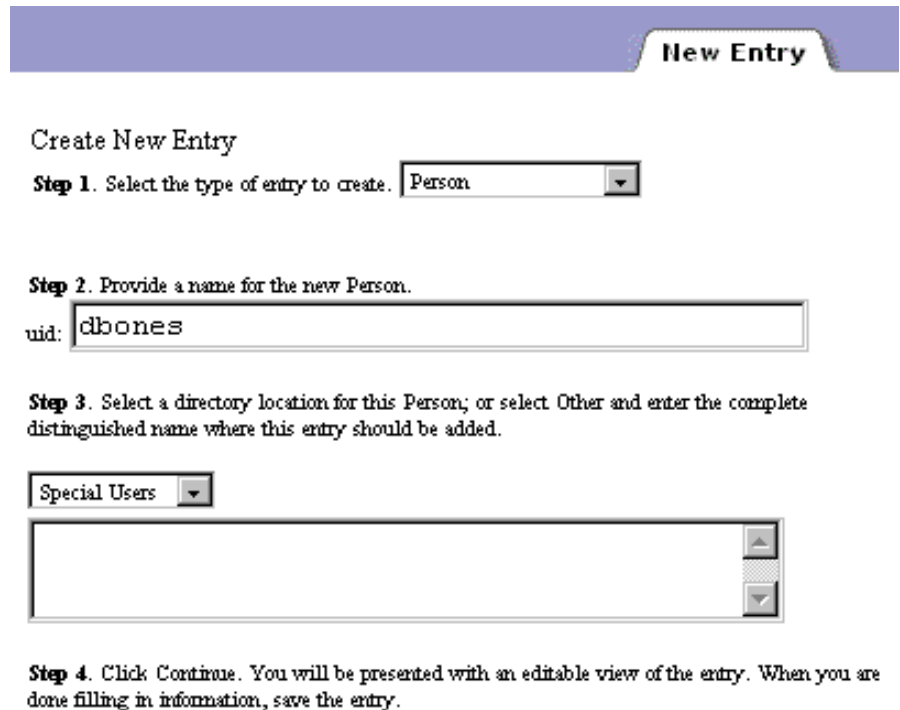
The template parameter is used to map the gateway's HTML templates for entry types to the directory server's LDAP object classes.

location

The location parameter is used to define 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 the entry will be formatted and the location in the directory tree where the entry will be created. Entry types for the default gateway appear in the pop-up menu gateway forms, for instance the Create New Entry form:

Figure 5-1 New Entry Form


Create New Entry

Step 1. Select the type of entry to create.

Step 2. Provide a name for the new Person.
uid:

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

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 will the new entry be created?

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

How will the new entry will be 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`.

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 the most commonly used directory branches only.

Mapping Entry Types to Locations

Each Entry type must be mapped to a location where that type of entry can be placed. The following example shows a mapping of locations and newtype parameters in `dsgw.conf`:

```
locationcountry"United States" "c=US#"
locationorg"This Organization" ""
locationgroups"Organizational Groups" "ou=Groups"
locationacct"Accounting" "ou=Accounting"
locationhr"Human Resources" "ou=Human Resources"
locationpay"Payroll" "ou=Payroll"
locationpd "Product Development" "ou=Product Development"
locationtest "Product Testing" "ou=Product Testing"

newtype orgperson"Person"cnacct hr pay pd test
newtype NTperson"NT Person"cnacct hr pay pd test
newtypegroupun"Group"cngroups
newtypeorgunit"Organizational Unit"ouorg
newtypeorg"Organization"ocountry
```

In the example, locations defined in the newtype parameter (such as `country`, `org`, or `groups`) correspond to handles defined in the location parameter. The friendly names (in quotes) in the third column indicate the choices that will appear in pull-down menus on gateway forms.

NOTE Location parameter definitions in `dsgw.conf` must be listed before newtype parameter definitions.

See “Entry Types (Object Classes)” on page 41 for more information.

Setting Up Organizational Units

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

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

Configuring DN Formats for Entry Types

The following sections describe how to configure entry type DNs depending upon the format.

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

The following sections describe the object classes and attributes contained by the 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.

Appendix B, “Gateway Directives” 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 the following example defines an Initials attribute for the 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 ideographic space for the non-breaking space () shown in the example.

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. The following example shows the steps required to delete the mobile phone attribute from the 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. The following example shows the steps required to add a template for a new object class, `siroePerson`. The new template adds two custom attributes—`dateOfBirth` and `preferredOS`—to the `inetOrgPerson` object class.

1. Copy the `display-orgperson.html` file and rename it as `display-siroeperson.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
<!-- siroe 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, siroeperson"
-->
```

4. Add the value for `siroeperson` 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="siroeperson">
```

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 46.

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

```
template siroeperson person inetorgperson siroeperson
```

This will instruct the gateway to display the `siroeperson` entry type according to the template defined for the `siroePerson` object class (`display-siroeperson.html`).

7. Update the directory server schema to include the `siroePerson` object class.
8. To allow users to add entries for `siroeperson` 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 “Considerations for Adding New Entry Types” on page 43, and “Extending Search Preferences” on page 59.

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 will instruct 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 will search for entries of this type.
5. Create a new search results form defining how the gateway will display search results for the new object class.

NOTE Modify an existing search result form to create a new search results form.

See “Adding Information to Search Results” on page 65 and “Removing Information From Search Results” on page 66.

Search Attributes, Filters, and Results

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

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

`dsgwsearchprefs.conf`

`dsgwsearchprefs.conf` specifies the preferences for searching object classes defined in the gateway. 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`

The `dsgwfilter.conf` file contains an entry for each search object defined in `dsgwsearchprefs.conf`. Each entry defines the following:

- Pattern to search for
- Delimiters for the search pattern
- LDAP filter for generating search results
- Text to use in describing search results for the pattern
- 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 searches on object classes defined for the gateway. The advanced search form allows users to also search in specific object class attributes and to specify a matching pattern. Figure 6-1, shows the matching patterns that can be selected in the advanced search form.

Figure 6-1 Entry Type Pop-up Menu for Advanced Search Form

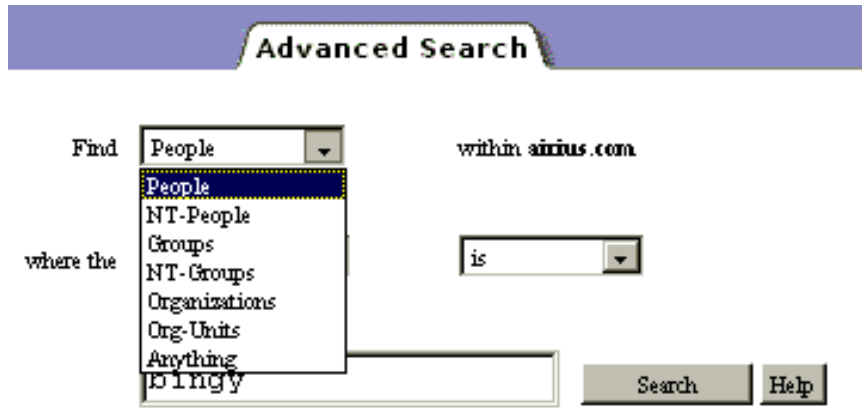


Figure 6-2 Attribute Pop-up Menu on Advanced Search Form

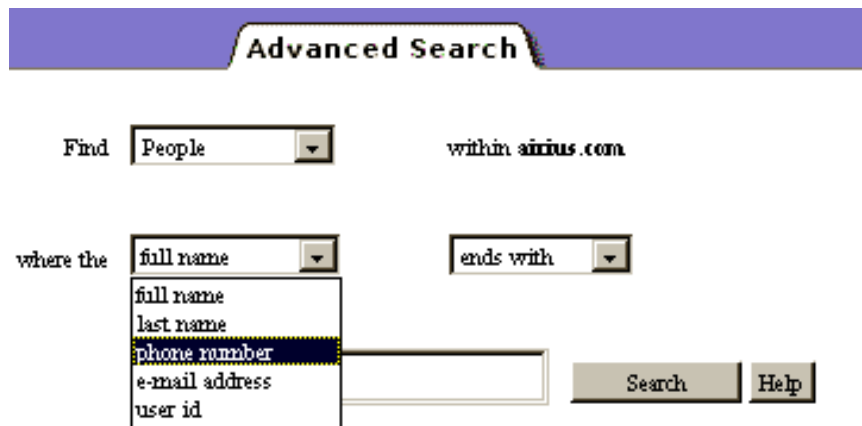
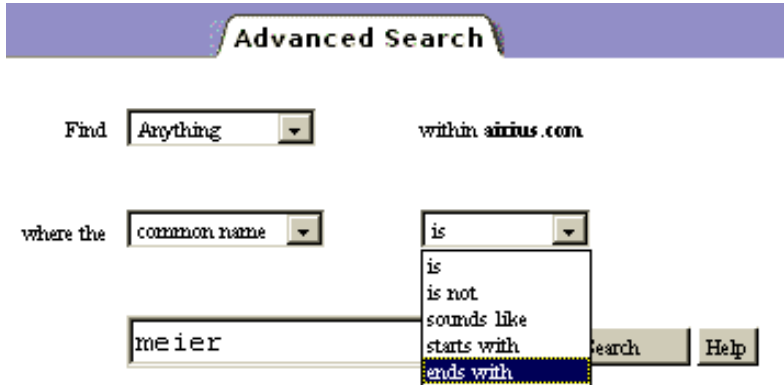


Figure 6-3 Matching Filter Pop-up on Advanced Search Form

Specifying Search Attributes for Person

The `dsgwsearchprefs.conf` syntax in the following example specifies that the `cn`, `sn`, `telephoneNumber`, `mail`, `uid`, and `title` attributes will be used in a search for person entries:

```

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 the example specifies how the LDAP attribute shown in the second column appears in the drop-down menu on the Advanced Search form.

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 indicates that the match type is valid for the associated attribute. A value of 0 indicates that the match type is not valid. In the example, the bit position for the telephone number attribute is set to 0, indicating that the directory server will 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
1	contains
2	ends with
3	starts with
4	sounds like
5	is not
6	is

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 does exact 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` and comment out the corresponding lines.

Adding Search Support for Additional Attributes

The boldfaced syntax in the following example specifies preferences for searching the `pagerTelephoneNumber` attribute.

```

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 will appear in the drop-down menu on the Advanced search form.
- The gateway will search the `pagerTelephoneNumber` attribute of all entries within the scope of the search.
- The gateway will look for values that contain, end with, start with, or are identical to the search string entered by the user. It will 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 specifying preferences for searching object class attributes and defining a filter to use in expressing search results.

Extending Search Preferences

The boldfaced syntax in the following example shows the introduction of a new object, `siroePerson`, and a new attribute, `dateOfBirth`, to the search preferences for the person object class.

```

People
""
"Search for":
"(&(objectClass=person)(objectClass=siroePerson)"
"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 syntax shown in the example:

- The gateway will include siroePerson entries in its search for persons.
- A “birthdate” field will appear on the advanced search form.
- The gateway will search the dateOfBirth attribute of all entries within the scope of the search.

The syntax added to dsgwfilter.conf in the following example defines the search values for the dateOfBirth attribute:

```

"[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

Adding a new search object that is not similar to existing search objects requires:

- A new entry in dsgwsearchprefs.conf
- A new entry in the dsgwfilter.conf
- A new search results template

The syntax in the following example specifies that the search definition for OrgRole will search the cn, l, roleOccupant, description, and telephoneNumber attributes:

```

"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 the above example specifies a list of filters for searching attributes in the `orgRole` object class.

This example creates 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"

```

Modifying search results forms is described in “Modifying Search Result Templates” on page 65.

Modifying Default Search Filters

The gateway uses `dsgwfilter.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.

Modify existing search filters in `dsgwfilter.conf` to support new user data patterns, instead of creating new filters.

Search Filters for User Data Patterns

This example shows typical search filter syntax for any search string containing the @ symbol. In this example, the gateway will respond 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 will then return 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).

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

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

Specifying a Search Filter for a New Object

The syntax in the following example allows users to search person entries by birthday or birth month:

```
[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 line `dateOfBirth=%v*))date of birth starts with` to the `dsgw-people` entry in `dsgwfilter.conf`, the gateway will also filter the `dateOfBirth` attribute for values that start with the supplied value (the filter). The gateway will return 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 will match many different strings.

Customizing Search Result Templates

The following sections describe how the gateway displays search results and contains procedures for customizing the gateway 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

Table 6-3 Default Search Results for Search Objects

Search object	Search result template used	Search results displayed
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

This example shows how additional information can be added to the search result by modifying the corresponding search result template.

```
<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>
```

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.

Customizing Graphics and Color

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

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

Appearance of Gateway Forms

The default gateway installed during directory server installation matches the standard appearance of iPlanet products. The gateway Interface Templates can be modified to customize the appearance of the following:

- Banner image
- Buttons
- Background color
- Text color
- Link color

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`.

Updating the Banner Image (`title.gif`)

The following sections describe how to change the dimensions of the banner image as well as how to change the banner image filename.

Changing Dimensions of Banner Image

The default iPlanet 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

Keep 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.

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
<code>stsearch_off.gif</code>	Used in the <code>maintitle.html</code> , <code>authtitle.html</code> , <code>csearchtitle.html</code> , and <code>newentrytitle.html</code> pages
<code>stsearch_on.gif</code>	Used on the <code>searchtitle.html</code> page
<code>adsearch_off.gif</code>	Used in the <code>maintitle.html</code> , <code>authtitle.html</code> , <code>searchtitle.html</code> , and <code>newentrytitle.html</code> pages
<code>adsearch_on.gif</code>	Used on the <code>csearchtitle.html</code> page
<code>newentry_off.gif</code>	Used in the <code>maintitle.html</code> , <code>authtitle.html</code> , <code>csearchtitle.html</code> , and <code>searchtitle.html</code> pages
<code>newentry_on.gif</code>	Used on the <code>newentrytitle.html</code> page
<code>authen_off.gif</code>	Used in the <code>maintitle.html</code> , <code>searchtitle.html</code> , <code>csearchtitle.html</code> , and <code>newentrytitle.html</code> pages
<code>authen_on.gif</code>	Used on the <code>authtitle.html</code> page

Updating Button Images

The default iPlanet 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	

Table 7-2 Files Controlling Appearance of Gateway Forms

To Change Colors on the	Edit	File Type
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 the following example shows the use of the COLORS directive to change text color to blue, background color to aqua, and the link color to red:

```
<!-- COLORS "attribute=color [attribute=color [...]]" -->
<!-- COLORS "TEXT=#0000FF BGCOLOR=#00FFFF LINK=#FF0000" >
DS_NEWENTRY_TYPE_BODY -->
```

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

Changing Table Colors

The following sections describe procedures for customizing the color of tables.

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	#800080
fuchsia	#FF00FF
green	#008000

Table 7-4 Sixteen Standard Colors and their Hexadecimal Values

Color name	Hexadecimal value
lime	#00FF00
olive	#808000
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:

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

so that the BGCOLOR value is an RGB color value in the form #RRGGBB or a standard color name. 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:

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

so that the BGCOLOR attribute specifies the RGB color value in the form #rrggbb or a standard color name representing the new color.

.conf Parameters

The `dsgw.conf` and `pb.conf` files are installed during directory server installation. Associated directives are described in Appendix B, “Gateway Directives.”

authlifetime

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.

For information on authenticating to the directory server using the gateway, see the on-line documentation that is available through the gateway.

Format

`authlifetime seconds`

Example

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

```
authlifetime 7200
```

baseurl

Specifies the hostname 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 hostname 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, searchbase is set to the directory's suffix.

Substitute the following hexadecimal values for the equal sign, space, and comma in the search base:

- use %3D instead of equal sign (=)
- use %20 instead of space ()
- use %2C instead of 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=siroe.com:

```
baseurl "ldaps://dirserver.siroe.com:636/o%3Dsiroe.com"
```

binddnfile

Specifies the location of the file where the bind DN and bind password are stored. This file should be stored separately of the .conf file for the gateway instance. The binddnfile is used to authenticate to the server for non-anonymous searching.

Format

binddnfile *binddnfilename*

Example

```
binddnfile /export/TEST/binddnfile
```

changeHTML

Used by the gateway to substitute ideographic space for non-breaking space (nbsp) in Asian character sets.

Format

changeHTML *nbsp_from nbsp_to charset*

Example

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

charset

Defines the default character set for communication with HTTP clients. The default is UTF-8 (Unicode), which supports all the characters in the iPlanet Directory Server. 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 release 4.0 of the gateway.

More information: "ignoreAccepCharsetFrom" on page 80

Format

```
charset charset
```

Example

```
charset UTF-8
```

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

For more information about how the 3.x and 4.0 releases of the gateway choose character sets for HTTP clients, see "Unicode and iPlanet Support for UTF-8" on page 29.

configdir

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 "../siroeconfig"
```

dirmgr

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. Use 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.

For information on authenticating as the directory manager from the gateway, see the on-line documentation that is available through the gateway.

Format

```
dirmgr "distinguished_name"
```

Example

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

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

gwnametrans

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

Format

```
gwnametrans "HTTP_redirect"
```

Example

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

html>dir

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 "/siroeconfig"
```

ignoreAccepCharsetFrom

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
```

More information: "charset" on page 77.

include

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

Format

```
include "config_file"
```

Example

```
include "../config/dsgw-110n.conf"
```

location

Defines the location choices selectable from the gateway when adding new entries. 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. For more information, see “location-suffix” on page 81.

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 pound sign (#), 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.

For more information, see “include” on page 80.

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=siroe.com#"
```

For a more complete example of the location parameter, see “Mapping Locations and Entry Types” on page 44.

location-suffix

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=siroe.com"
```

newtype

Defines the types of entries that can be added to the directory using the gateway. `newtype` 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
```

For a more complete example of the `newtype` parameter, see “Mapping Locations and Entry Types” on page 44.

NLS

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

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.

Format

```
securitypath "/usr/iplanet/servers/alias/cert.db"
```

Example

```
securitypath "/usr/iplanet/servers/alias/pb-cert.db"
```

template

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 template_name object_class
```

Example

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

```
template orgperson person inetorgperson
```

vcard-property

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. vCard properties that are currently mapped to LDAP attributes are:

- **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
```

Gateway Directives

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_.

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

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

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

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_ENTRYEND.

Arguments

None.

DS_ENTRYEND

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

Arguments

None.

DS_ATTRIBUTE

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 the following table:

Table B-1 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

Table B-1 DS_ATTRIBUTE: Display of syntax Argument

syntax type	description	display as
url	Display as a labeled URL	href (URL)

type=how-to-display. Renders the attribute on-screen in a particular format. Legal values described in the following table correspond roughly to HTML form element names.

Table B-2 DS_ATTRIBUTE: Display of type Argument

type	display
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 the following table:

Table B-3 DS_ATTRIBUTE: Display of options Argument

options	display
sort	Sort the attribute values
nolink	Do not attempt to display the attribute as a hyperlink
dntags	Applies only when using syntax=dn - tags are displayed when showing DNs. Normally they are not displayed.
dateonly	Applies only 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

Table B-3 DS_ATTRIBUTE: Display of options Argument

options	display
dnpicker	Applies only when using syntax=dn - embed delete checkboxes and Javascript array information. 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 from the LDAP server.

within=string-to-embed-in. For each value, outputs the text in string-to-embed-in, replacing all occurrences of the 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=Siroe Corporation, c=US", the output will be "James Doe, Siroe Corporation."

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

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

Arguments

value=value1,value2,...valueN. Specifies a list of object class values. For a template file to be used to display 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

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

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

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

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

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

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

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.

checkboxsubmit=*javascript*. Submit changes only if javascript expression is true.

Examples

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

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

DS_EDITASBUTTON

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

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

Arguments

None.

DS_CONFIRM_NEWPASSWORD

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 saves only the new password value if the two match. This directive must appear within a DS_ENTRYBEGIN...DS_ENTRY_END block.

Arguments

None.

DS_OLDPASSWORD

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

Displays a help button.

Arguments

topic=topic_name. Causes the iPlanet Help System to open the given topic name.

Example

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

DS_CLOSEBUTTON

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

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

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

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

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

Arguments

label=

template=

attr=

desc=

DS_BEGIN_DNSEARCHFORM

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

Arguments

None.

DS_ATTRVAL_SET

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

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 B-4 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 syntax 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 server using 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 Conditions marked with an asterisks (*) 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

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

Arguments

extrahtml

Examples

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

COLORS

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

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

Arguments

title-string

Example

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

ENDHTML

Emit </BODY></HTML> sequence

Arguments

None.

HELPBUTTON

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

Arguments

topic

Example

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

INCLUDE

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

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

Display a string that shows the result of the last `domodify` run. Note that this directive works only 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

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

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. However, conditionals marked with an asterisk (*) are supported.

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. Upper case words are variables. Lower case words are literals.

auth

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

```
authasrootdn = "true"
```

or

```
escapedbinddn = DN
```

lang

```
lang?context=CONTEXT[&file=FILE]
```

search

```
search?context=CONTEXT[&file=string]
```

(GETs only. No POSTs.)

csearch

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

`searchType = SEARCHTYPE`

`searchAttr = SEARCHATTR`

unauth

`unauth[?context=CONTEXT]`

dnedit

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

edit

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

(GETs only. No POSTs.)

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
```

newentry

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

```
entrytype = ET  
entryname = EN  
rdntag = RDNTAG  
selectLocation = SL  
dnsuffix = DNSUFFIX
```

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
```


Index

A

Additional CGI directory 18

Additional document directory 18

Advanced search

 attributes for 55

 example of pop-up menu 56

Attributes

 adding to object classes 47

 deleting from object classes 47

auth CGI 105

Authentication

 and non-anonymous searching 7

 as directory manager 23

authlifetime parameter 75

B

Banner files

 controlling appearance of gateway forms 40

Banner image 68

baseurl parameter 76

Bind DN 7, 13

Bind password 7, 13

binddnfile

 security for 8

binddnfile parameter 77

Button images

 on gateway forms 69

 updating 69

C

CGIs

- auth 105
- csearch 106
- dnedit 106
- dosearch 107
- edit 106
- lang 105
- newentry 107
- search 105
- specifying gateway configuration to 5
- unauth 106
- usage 105

changeHTML parameter 77

Chapter Single Template 75, 87, 105

Character sets 7

charset parameter 77

configdir parameter 78

csearch CGI 106

D

Default gateway

.conf file for 2

dsgw.conf 37

forms on 5

Directives

BODY 101

COLORS 101

context related 89

DS_ATTRIBUTE 90

DS_ATTRVAL_SET 99

DS_BEGIN_DNSEARCHFORM 99

DS_CLOSEBUTTON 98

DS_CONFIRM_NEWPASSWORD 97

DS_DELETEBUTTON 96

DS_DNEDITBUTTON 99

DS_EDITASBUTTON 97

DS_EDITBUTTON 96

- DS_EMIT_BASE_HREF 98
- DS_END_ENTRYFORM 98
- DS_ENTRYBEGIN 90
- DS_ENTRYEND 90
- DS_GATEWAY_VERSION 103
- DS_HELPBUTTON 98
- DS_LAST_OP_INFO 103
- DS_LOCATIONPOPUP 103
- DS_NEWPASSWORD 97
- DS_OBJECTCLASS 93
- DS_OLDPASSWORD 97
- DS_POSTEDVALUE 95
- DS_SAVEBUTTON 96
- DS_SEARCHDESC 95
- DS_SORTENTRIES 95
- DS_VIEW_SWITCHER 94
- ENDHTML 102
- entry related 89
- entry-related 46
- GCONTEXT 89
- HELPBUTTON 102
- IF/ ELSE/ ELIF/ ENDIF 99, 104
- in object class templates 46
- INCLUDE 102
- INCLUDECONFIG 102
- miscellaneous 101
- PCONTEXT 89
- TITLE 101
- used for changing colors on gateway forms 71

Directory Express

- .conf file for 2
- extended search result 4
- pb.conf 37
- search result 3
- search support for user ID 58

Directory manager

- authenticating as 23

- configuring DN for 22
- Directory server 4.0
 - gateways installed with 2
 - updating gateway with updatedsgw script 8
 - updating the gateway with changes to 14
- dirmgr parameter 79
- DN formats
 - configuring 45
 - modifying default DN format 45
 - UID-based 45
- dnedit CGI 106
- dosearch CGI 107
- dsgw-30.conf 14
- dsgwfilter.conf 37, 54
- dsgwmig script 13
- dsgwsearchprefs.conf 37, 53
- E
- edit CGI 106
- Entry types, see Object class templates
- G
- Gateway
 - and root suffix 16
 - compatibility with release 3.x 8
 - configuring for Netscape FastTrack Server 17
 - creating a new instance 20
 - HTTP server requirements, see also HTTP Configuration 2
 - installed with Directory Server 4.0 2
 - running in high-usage networks 15
 - support for multiple instances 5
 - URL for access, release 3.x 6
 - URL for access, release 4.0 6
 - what is 1
- Gateway CGIs 5
- Gateway clients, see HTTP clients
- Gateway cloning 12, 21
- Gateway configuration
 - .conf file 21

- configuring Communicator 4.x for preferred language 26
- configuring gateway clients 25
- configuring Navigator 3.01 for preferred language 26
- default port setting 22
- setting up directory manager 22
- setting suffix for adding entries 23
- setting up SSL support 24
- Gateway configurator
 - customizing Communicator's LDAP settings 27
- Gateway files
 - .conf file 36
 - .conf file configuration 21
 - banner files 40
 - See also Banner files
 - dsgw.conf 37
 - dsgwsearchprefs.conf 37
 - files controlling gateway functionality 35
 - invoking CGIs in script files 105
 - location of in release 3.x 12
 - location of in release 4.0 11
 - object class templates 38
 - See also Object class templates
 - pb.conf 37
 - script files 38
 - See also Script files
 - search configuration files 37
 - See also Search configuration files
 - search result templates 39
 - See also Search result templates
- Gateway forms
 - appearance of 67
 - banner image 68
 - button images 69
 - changing color of table headings 73
 - changing table background color 73
 - color schemes 70
 - files controlling color 70

- hexadecimal color values 72
 - on default gateway 5
- Gateway localization, see Localization
- Gateway migration, see Migration
- Gateway object classes 46
 - See also Object class templates
- Gateway performance on Netscape Administration Server 15
- Gateway planning, see Installation planning
- Gateway search filters, see Search filters
- Gateway search objects, see Search objects
- Gateway search preference, see Search preferences
- Gateway search results, see Search result templates
- Gateway security
 - protecting bind DN and password 13
- GET operations 5
- gwnametrans parameter 79
- H
 - htmlmdir parameter
 - Parameters
 - htmlmdir 79
- HTTP clients
 - configuring Communicator 4.x for preferred language 26
 - configuring Navigator 3.01 for preferred language 26
 - customizing Communicator's LDAP settings 27
 - displaying non-English alphabets
 - language support for 25
 - Unicode and Latin-1 character sets 25
- HTTP server
 - name trans mapping 16
- HTTP server configuration
 - add additional CGI directory (3.x) 19
 - add additional CGI directory (4.0) 19
 - add additional document directory (3.x) 19
 - add additional document directory (4.0) 18
 - and gateway root suffix 16
 - change permissions of cookie directory 20
 - name translation mapping 16

- HTTP server recommendations
 - for release 4.0 gateway 15
 - Netscape Administration Server performance 15

I

- ignoreAccetpCharsetFrom parameter 80
- include parameter
 - Parameters
 - include 80

- Installation planning 11

L

- lang CGI 105

Localization

- character sets 7
 - ideal character set for 24
 - setting up 24
- location parameter 42, 44, 80
- location-suffix parameter 81

M

Migration

- 3.x to 4.0 13
 - dsgw-30.conf 14
 - dsgwmig script 13
 - running the migration script 14

N

- Name translation mapping 16
- newentry CGI 107
- newtype parameter 42, 82
- NLS parameter 83
- Non-anonymous searching 7

O

Object class templates

- adding attributes to 47
- adding search support for 59
- adding template for a child of a parent class 49
- and directives 46
- and entry types 41
- and entry-related directives 46

- attributes 46
- configuring DN formats for 45
- considerations for adding new 43
- creating a new parent object class 50
- creating new search object 60
 - See also Search objects
- default object class templates 46
- default template files 38
- deleting attributes from 47
- extending 48
- mapping locations and entry types 44
- structure 88

P

Parameters

- authlifetime 75
- baseurl 76
- binddnfile 77
- changeHTML 77
- charset 77
- configdir 78
- dirmgr 79
- gwnametrans 79
- ignoreAcceptCharsetFrom 80
- location 42, 44, 80
- location-suffix 81
- newtype 42, 82
- NLS 83
- securitypath 83
- template 42, 84
- vcard-property 84

Port setting 22

POST operation

- using PCONTEXT 6

R

Root processes 13

Root suffix 16

S

Script files

- expressions for invoking CGIs in 105
- for Gateway forms 39

Search attributes 55

- adding 58
- adding search support 58
- search match types 58
- specifying 57

search CGI 105

Search configuration files

- dsgwfilter.conf 37, 54
- dsgwsearchprefs.conf 37, 53

Search filter

- for new object class 62

Search filters

- modifying 61
- user data patterns 62

Search objects

- creating 59
- creating new 60

Search preferences

- extending 59

Search result templates

- customizing 63
- default templates and related object classes 39
- modifying 65
- removing information from 66

Search results

- display of 63

Search scope

- base 55
- defining 54
- onelevel 55
- subtree 55

Search support for user ID in Directory Express 58

securitypath parameter 83

SSL support 24

Standard search page 5

T

template parameter 42, 84

U

unauth CGI 106

updatedsgw script 8, 14

updating title.gif 68

V

vcard-property parameter 84