



Sun Java™ System Directory Editor 1 2004Q4 Installation and Configuration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-1701

Copyright © 2004, 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

The OPEN LOOK and Sun(TM) Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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 © 2004, 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou des brevets supplémentaires ou des applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L'AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit peuvent être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Legato, le logo Legato, et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Who Should Use This Book	viii
Before You Read This Book	viii
How This Book Is Organized	ix
Conventions Used in This Book	x
Typographic Conventions	x
Symbols	xi
Mnemonics	xi
Default Paths and File Names	xi
Books in This Documentation Set	xii
Related Documentation	xii
Accessing Sun Resources Online	xii
Contacting Sun Technical Support	xiii
Related Third-Party Web Site References	xiii
Sun Welcomes Your Comments	xiv
Chapter 1 Before You Install	1
Supported Software Environments	1
Operating Systems	2
Application Servers	2
Browsers	3
Directory Servers	3
Recommended Directory Editor Configuration	3
Memory Requirements	4
Setup Task Flow	4
Prerequisite Tasks	4
What's Next?	5

Chapter 2 Installing Directory Editor	7
Before You Begin	7
Installation Steps	8
Step 1: Install an Application Server	8
Step 2: Install Directory Editor	10
Step 3: Install Directory Editor on an Application Server	13
Step 4: Specify the Startup Properties	28
Step 5: Specify the Managed Directory Properties	33
What's Next?	35
Chapter 3 Getting Started	37
Product Features	37
Logging In	38
Changing Your LDAP Password	39
Navigating Directory Editor	40
Browsing Directories	42
Using the Browse Button	44
Using Online Help	46
Chapter 4 Creating and Editing Objects	49
Creating Objects	49
Accessing the Create Page	50
Creating Users	50
Creating Groups	64
Creating Organizational Units	75
Creating Domains	80
Creating Localities	83
Editing Objects	86
Accessing an Edit Page	86
Enabling or Disabling Objects	91
Renaming Objects	93
Deleting Objects	95
Chapter 5 Working With Forms	97
Overview	97
How Do Forms Work?	99
Why Edit the Forms?	99
Accessing the Forms Page	101
Customizing Forms	103
Opening a Form	103
Working with the Forms Page	104
Performing Common Tasks	107

Creating New Forms	115
Deleting Customized Forms	118
Testing Your Form	118
Chapter 6 Searching Directories	119
Executing a Basic Search	120
Executing an Advanced Search	123
Executing a Filtered Search	129
Editing the Search Page Configuration	131
Accessing the Search Configuration Page	132
Specifying Default Settings	133
Configuring the Results Table	134
Specifying Search Types	137
Chapter 7 Configuring Directory Editor	141
Controlling User Access	141
Understanding Roles	142
Understanding Principals	143
Understanding Capabilities	143
Working with Roles	144
Accessing the Authorization Page	144
Creating Directory Editor Roles	145
Editing Roles	149
Deleting Roles	151
Working with Naming Attributes	151
Accessing the Naming Attributes Page	152
Creating New Object Class - Naming Attribute Mappings	153
Editing Naming Attributes	155
Deleting Selected Naming Attributes	156
Editing the Startup Properties	157
Editing the Managed Directory Properties	157
Chapter 8 Backing Up and Restoring Configurations	161
Accessing the Backup/Restore Page	162
Backing Up Your Configuration	162
Restoring Your Configuration	163
Chapter 9 Removing the Software	165
Removing the Software	165
Using the Directory Editor Uninstaller	165
Uninstalling Manually From a Windows Machine	166
Uninstalling Manually From a Unix Machine	167

Undeploying Directory Editor from Your Application Server	168
From Sun Application Servers	168
From Tomcat Application Servers	169
From WebLogic Application Servers	169
From WebSphere Application Servers	170
Chapter 10 Error Logging and Troubleshooting	171
Before You Call Technical Support	171
Obtaining Information About the Problem	172
Information in the Page	172
Information in the Logs	173
Common Problems and Solutions	177
Running in Non-Production Mode	177
java.lang.OutOfMemoryError	177
Failed to Save Startup Configuration Properties	178
java.security.AccessControlException	179
WebSphere is Extremely Slow During Startup	180
Changing the Object Class and Attributes Used for Configuration Objects	181
Short Session Timeout in Tomcat	182
de.war File Missing	182
Appendix A HTML Components Used to Define Directory Editor Forms	185
Overview	185
What Are HTML Components?	186
Component Classes	186
Basic Component Classes	186
Container Classes	187
Page Processor Requirements for HTML Components	187
Hidden Parameters	187
Component Subclasses	188
Naming Conventions	188
Data Types	189
Base Component Class	190
Basic Components	197
Appendix B Resources for Capability Configuration	207
Appendix C Improving Performance of Browse and Search Features	219
Introduction	220
Increasing Virtual List View Performance	220
Creating a VLV Index	220
From the Directory Server Console	221

From the Command Line	225
Creating Tree Indexes	226
Using the Virtual List View Anonymously	229
Index	1

List of Figures

Figure 2-1	Directory Editor License Agreement	10
Figure 2-2	Select Installation Directory Screen	11
Figure 2-3	Installation Summary Screen	12
Figure 2-4	Startup Properties Page	28
Figure 2-5	Managed Directory Page	32
Figure 3-1	Directory Editor Log In Page	38
Figure 3-2	Change My Directory Password Link	39
Figure 3-3	Edit Password Page	39
Figure 3-4	Directory Editor Home Page	40
Figure 3-5	Browse Tab	43
Figure 3-6	Example Browse Dialog Box	45
Figure 3-7	Completed Parent Entry Text Box	45
Figure 3-8	Online Help Window	46
Figure 4-1	Create Page	50
Figure 4-2	Quick Create Tab	51
Figure 4-3	Mike Miller's Common Attributes	53
Figure 4-4	Example Group Membership Tab	54
Figure 4-5	Example Location Information	55
Figure 4-6	Example Misc Attributes Tab	56
Figure 4-7	Example Extensions Tab	58
Figure 4-8	Example Mail User Tab	60
Figure 4-9	Example NT User Tab	62
Figure 4-10	Example UNIX User Tab	64
Figure 4-11	Create Page	65

Figure 4-12	Quick Create Tab	65
Figure 4-13	Static Group Tab	67
Figure 4-14	Search For Members to Add to the Group Search Page	67
Figure 4-15	Removing A Member from the Group	69
Figure 4-16	Importing Members	71
Figure 4-17	Dynamic Group Tab	72
Figure 4-18	Example Results Table	74
Figure 4-19	Quick Create Tab	76
Figure 4-20	Common Attributes Page	77
Figure 4-21	Locality Page	78
Figure 4-22	Quick Create Page	80
Figure 4-23	Other Page	82
Figure 4-24	Quick Create Tab	84
Figure 4-25	Locality Common Attributes Page	85
Figure 4-26	Enable the Checkbox in the Browse Tree	87
Figure 4-27	Click the Linked Name	87
Figure 4-28	Enable the Object in the Results Table	88
Figure 4-29	Example Edit an Object Page	88
Figure 4-30	Import/Export Tab	89
Figure 4-31	Specifying How Your Browser Should Open the File	90
Figure 4-32	Enable the Checkbox in the Browse Tree	92
Figure 4-33	Click the Object Name in the Results Table	92
Figure 4-34	Example Enable/Disable Pages	92
Figure 4-35	Enable the Checkbox in the Browse Tree	93
Figure 4-36	Enable the Object in the Results Table	93
Figure 4-37	Rename Page	94
Figure 4-38	Example Rename Page	94
Figure 4-39	Deleting Objects	95
Figure 5-1	Forms Tab	101
Figure 5-2	Action Column	102
Figure 5-3	Default Create Form	103
Figure 5-4	Tree View Display (Expanded)	104
Figure 5-5	Initial Property View: Create Form	105
Figure 5-6	Property View: Editing New Object Type Element	106
Figure 5-7	Identifying a Default Create Forms	107
Figure 5-8	New Form Page	108
Figure 5-9	Customizing Field Properties	109

Figure 5-10	Move Field Tab	112
Figure 5-11	Move Field Menu	113
Figure 5-12	Creating a New Field	114
Figure 5-13	Creating a New Form	115
Figure 5-14	Adding Classes to a Form	117
Figure 5-15	Specify Properties for Each Node in the Tree	117
Figure 6-1	Search Results	120
Figure 6-2	Search Results	121
Figure 6-3	Advanced Search Tab	123
Figure 6-4	Selecting Object Classes	124
Figure 6-5	Example 1 Search Results	127
Figure 6-6	Example 2 Search Results	128
Figure 6-7	Filtered Search tab	129
Figure 6-8	Example Filtered Search Results	130
Figure 6-9	Search Configuration Page	132
Figure 6-10	Results Table Page	134
Figure 6-11	Search Types Page	137
Figure 7-1	Authorization Page	144
Figure 7-2	Role Properties Tab	145
Figure 7-3	Principals Tab	146
Figure 7-4	Search for Principals Page	146
Figure 7-5	Adding Objects to the Principals Set	147
Figure 7-6	Principal Table	147
Figure 7-7	Capabilities Tab	148
Figure 7-8	New Role Added to the Roles Table	149
Figure 7-9	Click the Checkbox	149
Figure 7-10	Edit Page	150
Figure 7-11	Click the Checkbox	151
Figure 7-12	Naming Attributes Page	152
Figure 7-13	New Naming Attributes Page	153
Figure 7-14	New Object Class and Naming Attribute Added to the Table	154
Figure 7-15	Updated Table	155
Figure 7-16	Editing the Naming Attributes	155
Figure 7-17	Click the Checkbox	156
Figure 7-18	Search for Principals button	157
Figure 7-19	Search for Principals Page	158
Figure 7-20	Adding Objects to the Principals Set	158

Figure 7-21	New Manager Principals List	159
Figure 8-1	Backup/Restore Page	162
Figure 8-2	Specifying How Your Browser Should Open the File	163
Figure 10-1	Enabling Audit Logging	176
Figure C-1	Creating a vlvSearch Object	221
Figure C-2	Selecting the vlvSearch Object Class	222
Figure C-3	Specifying the Object Class Attributes	222
Figure C-4	Changing the Naming Attributes	223
Figure C-5	Creating a vlvIndex Object Class	224

List of Tables

Table 1	Typographic Conventions	x
Table 2	Symbol Conventions	xi
Table 3	Default Paths and File Names	xi
Table 4	Books in This Documentation Set	xii
Table 4-1	Mail User Attributes	59
Table 4-2	NT User Attributes	61
Table 4-3	Unix User Attributes	63
Table 5-1	Directory Editor Forms	97
Table 5-2	Field Properties	110
Table 6-1	Example Search Parameters	126
Table 6-2	Example Search Parameters	127
Table A-1	Hidden Parameter Names	187
Table A-2	Component Properties Data Types	189
Table A-3	Command Property Values	195
Table B-1	Resources Available for Role Configuration	208
Table C-1	Settings When the Tree is Not Filtering	226
Table C-2	Settings to When the Tree is Filtering Groups	227
Table C-3	Settings When Tree is Filtering Users	227
Table C-4	Settings When the Tree is Filtering Users and the testuser String	228
Table C-5	Settings When the Tree is Filtering the testuser String	228

Preface

This publication explains how to install and configure Sun Java™ System Directory Editor for use in a production environment.

For the latest information about the features and functionality in this release of Directory Editor, please see the online release notes at:

<http://docs.sun.com/db/prod/sjs.diredit>

NOTE User interfaces depicted in this document are subject to change in future versions of the product.

This Preface contains the following information:

- “Who Should Use This Book” on page viii
- “Before You Read This Book” on page viii
- “How This Book Is Organized” on page ix
- “Conventions Used in This Book” on page x
- “Books in This Documentation Set” on page xii
- “Related Documentation” on page xii
- “Accessing Sun Resources Online” on page xii
- “Contacting Sun Technical Support” on page xiii
- “Related Third-Party Web Site References” on page xiii
- “Sun Welcomes Your Comments” on page xiv

Who Should Use This Book

This *Installation and Configuration Guide* is intended for use by administrators, systems engineers, and professional services engineers who will use Directory Editor to create and manage directory data.

You should already be familiar with

- Creating and managing directory, organization, group, user, and container data
- Configuring and operating Sun Java™ System Directory Server
- Lightweight Directory Access Protocol (LDAP)
- Java technology
- Extensible Markup Language (XML)

Before You Read This Book

The *Sun™ Java System Directory Editor Release Notes* contain the latest information about the product — including information that may supersede instructions provided in this book. Be sure you read these Release Notes before attempting any procedures described in this book.

Because Sun Java™ System Directory Server is used as the data store in an Directory Editor deployment, you should be familiar with the documentation provided with that product. Directory Server documentation can be accessed online at

http://docs.sun.com/coll/DirectoryServer_04q2

How This Book Is Organized

The *Sun Java™ System Directory Editor Installation and Configuration Guide* is organized into the following chapters:

- Chapter 1, “Before You Install” provides important information to help you prepare to install and configure Directory Editor.
- Chapter 2, “Installing Directory Editor” provides instructions for installing Directory Editor on your system.
- Chapter 3, “Getting Started” provides an overview of the Directory Editor’s graphical user interface and major features.
- Chapter 4, “Creating and Editing Objects” explains how to create and edit objects in your directories.
- Chapter 5, “Working With Forms” explains how to customize the default Directory Editor forms and how to create new forms for your site.
- Chapter 6, “Searching Directories” explains how to locate different objects in the directory. Directory Editor enables you to perform basic, advanced, and filtered searches.
- Chapter 7, “Configuring Directory Editor” explains how to configure authorization roles and create a secure site.
- Chapter 8, “Backing Up and Restoring Configurations” explains how to back-up and restore configuration data.
- Chapter 9, “Removing the Software” explains how to uninstall Directory Editor.
- Chapter 10, “Error Logging and Troubleshooting” explains how to work with Directory Editor error logs and how to troubleshoot problems you may encounter as you use the product.
- Appendix A, “HTML Components Used to Define Directory Editor Forms” describes the HTML components you use to customize Directory Editor forms.
- Appendix B, “Resources for Capability Configuration” describes the resources that are available for configuring roles.

Conventions Used in This Book

The tables in this section describe the conventions used in this book. The information is organized as follows:

- “Typographic Conventions” on page x
- “Symbols” on page xi
- “Mnemonics” on page xi
- “Default Paths and File Names” on page xi

Typographic Conventions

The following table describes the typographic conventions used in this book.

Table 1 Typographic Conventions

Typeface	Meaning	Examples
AaBbCc123 (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, on-screen computer output, sample code.	Edit your <code>.login</code> file. The <code>http://docs.sun.com</code> web site enables you to access Sun technical documentation online.
AaBbCc123 (Monospace bold)	What you type, when contrasted with on-screen computer output.	<code>% su</code> Password:
<i>AaBbCc123</i> (Italic)	Book titles, new terms, words to be emphasized. A placeholder in a command or path name to be replaced with a real name or value.	Refer to the <i>Sun Java™ System Directory Editor 1 2004Q4 Release Notes</i> . <i>Do not</i> save the file. The file is located in the <code><install-dir>/bin</code> directory.

Symbols

The following table describes the symbol conventions used in this book.

Table 2 Symbol Conventions

Symbol	Description	Example	Meaning
[]	Contains optional command options.	ls [-1]	The -1 option is not required.
{ }	Contains a set of choices for a required command option.	-d {y n}	The -d option requires that you use either the <i>y</i> argument or the <i>n</i> argument.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
>	Indicates menu item selection in a graphical user interface.	File > New > Templates	From the File menu, choose New. From the New submenu, choose Templates.

Mnemonics

Directory Editor uses *mnemonics* (underlined letters) throughout the user interface to give you quicker options for performing certain tasks. You type the underlined letter and press the Alt key simultaneously to perform the task. Mnemonics are not case sensitive.

Default Paths and File Names

The following table describes the default paths and file names used in this book.

Table 3 Default Paths and File Names

Term	Description
<installation_root>/<machine_name>/logs	Default path to the Directory Editor central logs (on Unix)
C:<installation_root>-<version_number>\webapps	Default installation directory for the de.war file (on Windows)

Books in This Documentation Set

The following table summarizes the books included in the Directory Editor documentation set.

Table 4 Books in This Documentation Set

Book Title	Description
<i>Sun Java System Directory Editor Installation and Configuration Guide</i>	Describes how to install and configure Directory Editor for use in a production environment.
<i>Sun Java System Directory Editor Release Notes</i>	Available after the product is released. Contains last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Related Documentation

Because you will be working with Sun™ Java System Directory Server, you may need to refer to the Directory Server product documentation. You can access this documentation from the following location:

http://docs.sun.com/coll/DirectoryServer_04q2

The <http://docs.sun.com> web site enables you to access Sun technical documentation online. You can browse the archive or search for a specific book, title, or subject.

Accessing Sun Resources Online

For product downloads, professional services, patches and support, and additional developer information, go to the following:

- **Developer Information**
<http://developers.sun.com/prodtech/index.html>
- **Download Center**
<http://www.sun.com/software/download/>
- **Product Data Sheets**
<http://www.sun.com/software/>

- **Product Documentation Online**
<http://docs.sun.com>
- **Product Support and Status**
<http://www.sun.com/service/support/software/>
- **Professional Services**
<http://www.sun.com/service/sunps/sunone/index.html>
- **Sun Enterprise Services, Solaris Patches, and Support**
<http://sunsolve.sun.com>
- **Support and Training**
<http://www.sun.com/supporttraining/>

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in the product documentation, go to the Sun Support and Services website located at:

<http://www.sun.com/service/contacting>

Related Third-Party Web Site References

This publication references the following, third-party web site for information about installing the Apache Tomcat software:

<http://jakarta.apache.org/tomcat/>

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to <http://docs.sun.com> and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document.

For example, the title of this book is *Sun™ Java System Directory Editor Installation and Configuration Guide*, and the part number is 819-1701.

Before You Install

Use the information provided in this chapter to prepare for the installation of Sun™ Java System Directory Editor (Directory Editor). This information is organized as follows:

- “Supported Software Environments” on page 1
- “Recommended Directory Editor Configuration” on page 3
- “Memory Requirements” on page 4
- “Setup Task Flow” on page 4
- “Prerequisite Tasks” on page 4
- “What’s Next?” on page 5

Supported Software Environments

This section lists the software and environments that are compatible with Directory Editor:

- “Operating Systems” on page 2
- “Application Servers” on page 2
- “Browsers” on page 3
- “Directory Servers” on page 3

NOTE Because software product developers frequently ship new versions, updates, and fixes to their software, the information published here changes often. Review the product release notes for updates before proceeding with installation.

Operating Systems

Directory Editor is compatible with the following operating systems:

- Solaris 8 or later
- AIX 4.3.3 or later
- Microsoft Windows 2000 SP3 or later
- Microsoft Windows 2003
- Red Hat Linux

Application Servers

Directory Editor supports any application server that supports Servlet 2.2 or later, but Directory Editor operates most efficiently with the following application servers:

- Sun Java™ System Application Server 8 or later
- Sun ONE Application Server 7 or later
- Apache Tomcat 4.1.x or later
- BEA WebLogic 8.1 or later
- IBM WebSphere 4 or later

NOTE You must configure your application server's policy file to give Directory Editor permission to access certain system resources. Instructions for editing the policy file for the different application servers are provided in "Step 3: Install Directory Editor on an Application Server" on page 13.

Browsers

Directory Editor is compatible with the following browsers:

- Firefox 1.0
- Netscape 7.0 and later
- Mozilla 1.4 and later
- Microsoft Internet Explorer (IE) 5.5 and later

Directory Servers

Directory Editor works with Sun Java™ System Directory Server 5.1 or Sun ONE Directory Server to provide you with enhanced functionality and optimal performance.

Recommended Directory Editor Configuration

When configuring Directory Editor, the following configurations are recommended for optimum performance and support:

- Sun Solaris 9, running one of the following application servers:
 - Sun ONE Application Server 7.0
 - WebLogic 7.2
- Microsoft Windows 2000 SP4, running one of the following application servers:
 - Sun ONE Application Server 7.0
 - Apache Tomcat 5.0
 - WebLogic 7.2
 - WebLogic 8.1
 - WebSphere 5.0
- IBM AIX 4.3.3, running WebSphere 5.0
- IBM AIX 5.2, running WebSphere 5.1

This configuration list is based on input from customer configurations, support, professional services, and pre-sales.

Memory Requirements

You can determine memory requirements and set values in your application server's JVM (Java Virtual Machine). Add the maximum and minimum heap size to the Java command line, for example:

```
java -Xmx128M -Xms128M
```

NOTE For the best performance, set these values to the same size.

Setup Task Flow

Your installation and configuration task flow will depend somewhat on which application server and database server you decide to use. In general, you will perform the following tasks:

- Perform the prerequisite tasks (described in the next section)
- Install the Directory Editor `de.war` file
- Configure Directory Editor

Depending on your server types; you may combine steps, perform them in a different order, or omit them altogether.

Prerequisite Tasks

Before you install the software, you must perform the following tasks:

- Install and configure the Java 1.4.2 (or later) Java Development Kit (J2SDK) if a J2SDK is not provided by your application server.
- Install and configure an application server.
- Install Directory Server.

NOTE For detailed information about installing and configuring Sun Java™ System Directory Server, consult the *Sun Java™ System Directory Server 5 2004Q2 Installation and Migration Guide*.

- Select a *configuration directory* and a *managed directory*.
 - The Directory Editor configuration directory is a directory where you store the product's configuration information.

This directory does not have to be a configuration directory as defined by the *Sun ONE Server Console 5.x Server Management Guide*.
 - The managed directory is the directory server serving as the source of data being managed by Directory Editor.

NOTE You can use the same directory for both the configuration and the managed directories.

- If you are *upgrading* or *reinstalling* Directory Editor, you must uninstall the older product version first.
 - a. If necessary, save your configuration as described in “Backing Up Your Configuration” on page 162.
 - b. Run the Directory Editor uninstaller as described in “Using the Directory Editor Uninstaller” on page 165.
 - c. Install (or reinstall) Directory Editor as described in Chapter 2, “Installing Directory Editor.”

What's Next?

Continue to Chapter 2, “Installing Directory Editor” to install and configure Directory Editor.

What's Next?

Installing Directory Editor

Use the information and procedures described in this chapter to install Directory Editor for use with different application servers. This chapter is organized as follows:

- “Before You Begin” on page 7
- “Installation Steps” on page 8
- “What’s Next?” on page 35

Before You Begin

Before starting the Directory Editor installation process,

- Ensure that your application server or servlet container uses the Java 1.4.2 J2SDK (or later)
- Ensure you have the following information at-hand:
 - If your application server is already installed, note the installation location
 - Your login and password for the Directory Server

NOTE You must have an account with read and write access for the directory server you will use to store the Directory Editor configuration.

Installation Steps

Use the following steps to install Directory Editor:

1. Install an application server (if it is not already installed on your machine).
2. Install the Directory Editor software.
3. Install and deploy Directory Editor on your application server.
4. Set the Startup Properties.
5. Set the Managed Directory.
6. Log-in.

Step 1: Install an Application Server

You must have a Java compiler and a Java Virtual Machine installed on your machine to run the Java classes that perform actions in Directory Editor. Both items are available with the Java 2 Software Developer's Kit (J2SDK). (The JRE packages do not provide a compiler.)

You can download the J2SDK from <http://java.sun.com>, and follow the installation instructions provided with the product *or* many application servers bundle a J2SDK with their installation.

CAUTION Directory Editor requires access to some privileged operations that the application server may not allow by default. Edit the security policy file for your application server and grant access to certain actions, as follows:

1. Open the `server.policy` file.
2. Add the following lines to the end of the block granting access to “all remaining code:”

```
permission javax.security.auth.AuthPermission
"getLoginConfiguration";

permission javax.security.auth.AuthPermission
"setLoginConfiguration";

permission javax.security.auth.AuthPermission
"createLoginContext.SunDirectoryLogin";

permission javax.security.auth.AuthPermission
"modifyPrincipals";
```

3. Save the file.
 4. Restart the application server.
-

NOTE

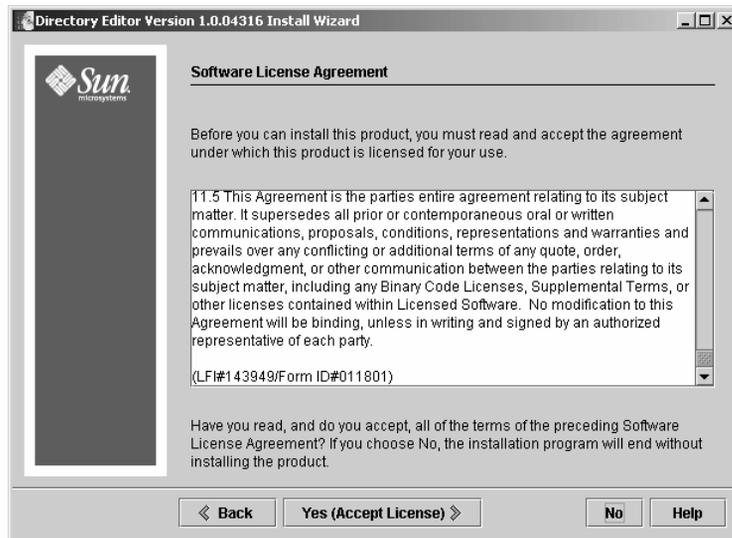
- The J2SDK version that is shipped with an application server is always preferred to any other J2SDKs installed on your server.
- When installing the J2SDK, you can conserve space on your machine if you do not install the Java documentation, examples, or source code.
- Be aware that adding `JAVA_HOME` to your list of system environment variables benefits the application server, but it may affect other applications.

Step 2: Install Directory Editor

Use the following instructions to download and install the Directory Editor `de.war` file.

1. Download the `DE12004Q4SP1.zip` file from the Sun website to a safe location in your deployment directory. This zip file contains the following files:
 - o Readme file
 - o Third-party license file
 - o Directory Editor license file
 - o Directory Editor installer
2. Extract these files from the `DE12004Q4SP1.zip` file.
3. From a command window, type `java de` to launch the Directory Editor installer.
4. When the Welcome screen is displayed, click Next to open the Directory Editor Software License Agreement (shown in the following figure).

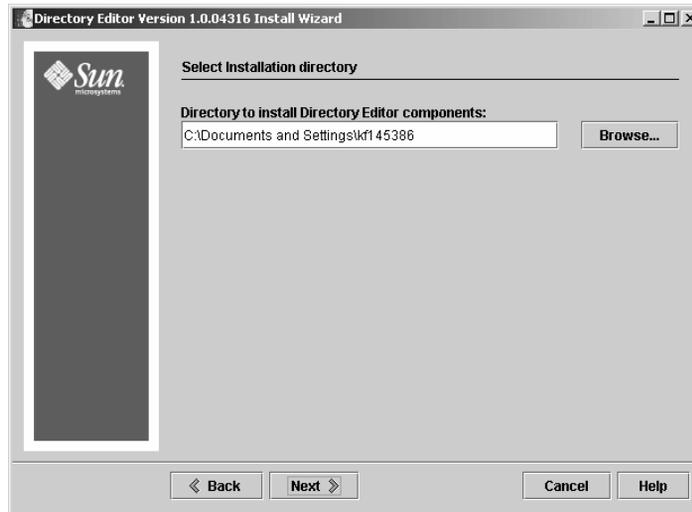
Figure 2-1 Directory Editor License Agreement



- o Click Yes (Accept License) if you agree to the terms.
- o Click No to cancel the installation.

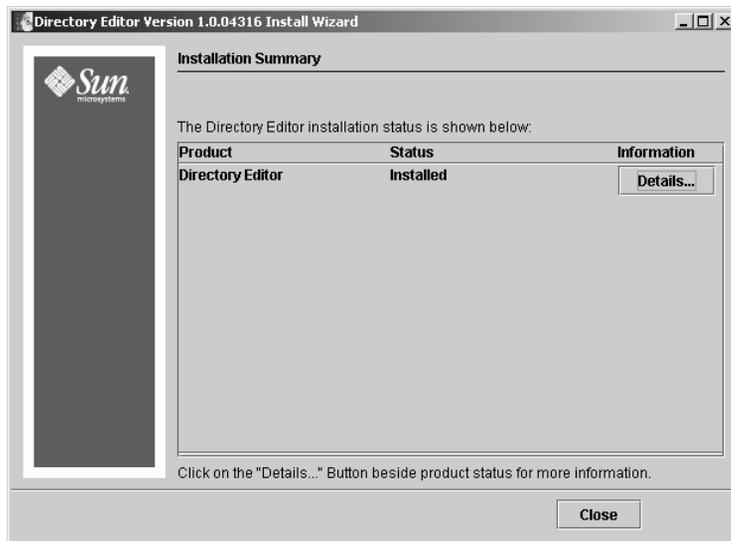
5. If you accept the license agreement, the Select Installation Directory screen displays. Enter the directory and path where you want to install the product components (or click Browse to locate and select a directory).

Figure 2-2 Select Installation Directory Screen



6. Click Next and the Ready to Install screen is displayed to provide information about the product.
7. When you ready, click the Install Now button.

An Installing... screen displays briefly during the installation process, and then the Installation Summary screen (Figure 2-3) is displayed to report the installation status.

Figure 2-3 Installation Summary Screen

If necessary, click the Details button for additional information.

8. Click Close to exit the installer.

Next, you must install and deploy the `de.war` file on your application server. Continue to “Step 3: Install Directory Editor on an Application Server” on page 13 for instructions.

Step 3: Install Directory Editor on an Application Server

After downloading and installing the Directory Editor `de.war` file, use the instructions provided in this section to install and deploy the software on your application server.

The information is organized as follows:

- “Installing Directory Editor on a Sun Application Server” on page 13
- “Installing Directory Editor for Tomcat 5.0.x” on page 17
- “Installing Directory Editor for WebLogic” on page 24
- “Installing Directory Editor for WebSphere 5.1” on page 26

Installing Directory Editor on a Sun Application Server

This section provides instructions for installing Directory Editor on a Sun Application Server, which consists of the following steps:

- “Installing the Sun Application Server Software” on page 13
- “Configuring the Sun Application Server’s Policy File” on page 14
- “Deploying Directory Editor into a Sun Applications Server” on page 16

Installing the Sun Application Server Software

NOTE The following information is provided for general reference only. For detailed installation instructions, refer to the installation instructions provided with your Sun Application Server or download the product documentation from <http://docs.sun.com>.

You may need to perform one or more of these general steps when installing the software:

- Use the Sun Application Server’s Typical installation.
- Specify a location for the installation directory.
- Specify an administrator name and password for Application Server administration.

Configuring the Sun Application Server's Policy File

You must configure the application server's policy file to give Directory Editor permission to access the application server.

1. Define a `de.home` variable as follows:
 - a. Launch and log into the Sun ONE Admin Console.
 - b. On the left side of the Console, click the folder name for the Application Server instance on which Directory Editor will be installed.

NOTE The default folder name is usually *server1*.

- c. On the left side of the Console, select the JVM settings tab, and then click the JVM Options link.
- d. Add the following JVM option for `de.home`:

```
-Dde.home=<SunONEHome>\<domain_name>\<App_Server_Instance>\
applications\j2ee-modules\<idm_1>
```

For example:

```
-Dde.home=-Dde.home=/opt/SUNWappserver7/domain1/server1/
applications/j2ee-modules/de_1
```

2. Add the following line to the `<App Server Home>/domains/<Domain Name>/config/server.policy` file (see Code Example 2-1):

```
grant codeBase "file:${de.home}/-" {
permission java.util.PropertyPermission "*", "read,write";
permission java.io.FilePermission "<<ALL FILES>>", "execute";
};
```

Code Example 2-1 Example Sun Application Server Policy File

```

/* Copyright (c) 2002 Sun Microsystems, Inc. All rights reserved. */
/* Use is subject to license terms. */

//System Code Permissions

grant { permission java.security.AllPermission; };

grant codeBase "file:${de.home}/-" {
    permission java.util.PropertyPermission "*", "read,write";
    permission java.io.FilePermission "<<ALL FILES>>", "execute";
};

//Web Application Permissions
grant {
    permission java.lang.RuntimePermission "accessClassInPackage.sun.io";
    permission java.lang.RuntimePermission "createClassLoader";
    permission java.lang.RuntimePermission "accessDeclaredMembers";
    permission com.waveset.repository.test.testConcurrentLocking "read";
    permission java.net.SocketPermission "*", "connect,resolve";
    permission java.io.FilePermission "/export/logs/trace.log", "read,write";
    permission java.util.PropertyPermission "trace.file", "read";
    permission java.util.PropertyPermission "trace.destination", "read";
    permission java.util.PropertyPermission "trace.enabled", "read";
    permission javax.security.auth.AuthPermission "getLoginConfiguration";
    permission javax.security.auth.AuthPermission "setLoginConfiguration";
    permission javax.security.auth.AuthPermission "createLoginContext.SunDirectoryLogin";
    permission javax.security.auth.AuthPermission "modifyPrincipals";

};

// Core server classes get all permissions by default
grant codeBase "file:${com.sun.aas.installRoot}/lib/-" {
    permission java.security.AllPermission;
};

// iMQ classes get all permissions by default
grant codeBase "file:${com.sun.aas.imqLib}/-" {
    permission java.security.AllPermission;
};

```

```

// Web Services classes get all permissions by default
grant codeBase "file:${com.sun.aas.webServicesLib}/-" {
    permission java.security.AllPermission;
};

// Basic set of required permissions granted to all remaining code
grant {
    permission java.lang.RuntimePermission "loadLibrary.*";
    permission java.lang.RuntimePermission "queuePrintJob";
    permission java.net.SocketPermission "*" , "connect";
    permission java.io.FilePermission "<<ALL FILES>>" , "read,write";
    permission java.util.PropertyPermission "*" , "read";

    permission java.lang.RuntimePermission"modifyThreadGroup";
};
// Following grant block is only required by Connectors. If Connectors
// are not in use the recommendation is to remove this grant.
grant {
    permission javax.security.auth.PrivateCredentialPermission
"javax.resource.spi.security.PasswordCredential com.sun.enterprise.security.PrincipalImpl
\"ANONYMOUS\"" , "read";
};
// Following grant block is only required for Reflection. If Reflection
// is not in use the recommendation is to remove this section.
grant {
    permission java.lang.RuntimePermission "accessDeclaredMembers";
};

```

Deploying Directory Editor into a Sun Applications Server

Use the following steps to deploy Directory Editor into the Sun Application Server:

1. Launch the Sun Admin Console and log in.
2. Navigate to the Web Apps folder icon in the left panel (for example, select App Server Instance >server1 >Applications >Web Apps) and click the folder icon.
3. In the right panel, select the Deploy tab.
4. Enter the file path for the de.war file, and then click OK.
5. When prompted, set both the Web Application Name and the Context Root to de, and then click OK.

6. If you used the defaults at install time and you are not using a Web Server, open your browser and type `http://localhost:<port_number>/de/` into the URL field. (The port number will vary.)

NOTE If you are running your web browser on a host other than the application server you may have to change or adjust the host name in the URL.

The Directory Editor Startup Properties page is displayed. Continue to “Step 4: Specify the Startup Properties” on page 28 for instructions.

Installing Directory Editor for Tomcat 5.0.x

This section provides general instructions for installing an Apache Tomcat application server, and then explains two methods for installing Directory Editor on an Tomcat application server. The information is organized as follows:

- “Installing Tomcat” on page 17
- “Installing Directory Editor Using Tomcat Manager” on page 22
- “Installing Directory Editor Manually” on page 23

Installing Tomcat

NOTE The instructions provided in this section are provided for general reference only. For detailed installation instructions, consult the Apache website (<http://jakarta.apache.org/tomcat/>) or the reference information provided for the application server software.

If you are installing the application server from the Tomcat installer

1. Download and unpack the Tomcat installation bundle.
2. Decide where to install your Tomcat installation.
3. Specify to start Tomcat as a service (*on Windows only*).
4. Select a port (default is *8080*).

5. Modify the Tomcat start-up script as follows:

- **On UNIX:** Open the `$CATALINA_HOME/bin` directory and add the following lines to the top of the `setclasspath.sh` file:

```
JAVA_HOME=<J2SDK location>
```

```
export JAVA_HOME
```

- **On Windows:** Open the `$CATALINA_HOME/bin` directory and add the following lines to the top of the `setclasspath.bat` file:

```
SET JAVA_HOME=<J2SDK location>
```

Configuring the Policy File

If you will be running Tomcat with the Security Manager turned on, you must configure the application server's `catalina.policy` file (located in `$Appserver_home\conf\catalina.policy`) to give Directory Editor permission to access the application server. Add the following line to the bottom of the file (see Code Example 2-3):

```
grant codeBase "file:${catalina.home}/webapps/de/-" {
  permission java.security.AllPermission;
};
```

Code Example 2-2 Example `catalina.policy` File

```
/ =====
// catalina.corepolicy - Security Policy Permissions for Tomcat 5
//
// This file contains a default set of security policies to be enforced (by the JVM)
// when Catalina is executed with the "-security" option. In addition to the permissions
// granted here, the following additional permissions are granted to the codebase specific
// to each web application:
//
// * Read access to the document root directory
//
// $Id: catalina.policy,v 1.11 2004/03/02 12:36:22 renmm Exp $
// =====
```

Code Example 2-3 Example catalina.policy File

```

// These permissions apply to javac grant codeBase "file:${java.home}/lib/-"
{permission java.security.AllPermission;
};
// These permissions apply to all shared system extensions grant codeBase
"file:${java.home}/jre/lib/ext/-" {permission java.security.AllPermission;
};
// These permissions apply to javac when ${java.home} points at $JAVA_HOME/jre grant
codeBase "file:${java.home}/../lib/-" {permission java.security.AllPermission;
};
// These permissions apply to all shared system extensions when
// ${java.home} points at $JAVA_HOME/jre grant codeBase "file:${java.home}/lib/ext/-"
{permission java.security.AllPermission;
};

// ===== SYSTEM CODE PERMISSIONS =====

// ===== CATALINA CODE PERMISSIONS =====
// These permissions apply to the launcher code grant codeBase
"file:${catalina.home}/bin/commons-launcher.jar" {permission java.security.AllPermission;
};

// These permissions apply to the daemon code grant codeBase
"file:${catalina.home}/bin/commons-daemon.jar" {permission java.security.AllPermission;
};
// These permissions apply to the commons-logging API grant codeBase
"file:${catalina.home}/bin/commons-logging-api.jar"
{permission java.security.AllPermission;
};
// These permissions apply to the server startup code grant codeBase
"file:${catalina.home}/bin/bootstrap.jar" {permission java.security.AllPermission;
};
// These permissions apply to the JMX server grant codeBase
"file:${catalina.home}/bin/jmx.jar" {permission java.security.AllPermission;
};

// These permissions apply to the servlet API classes and those that are shared across all
// class loaders located in the "common" directory grant codeBase
"file:${catalina.home}/common/-" {permission java.security.AllPermission;
};

```

```
// These permissions apply to the container's core code, plus any additional
// libraries installed in the "server" directory
grant codeBase "file:${catalina.home}/server/-" {permission java.security.AllPermission;
};
// ===== WEB APPLICATION PERMISSIONS =====
// These permissions are granted by default to all web applications
// In addition, a web application will be given a read FilePermission and JndiPermission
// for all files and directories in its document root.grant {
    // Required for JNDI lookup of named JDBC DataSource's and
    //javamail named MimePart DataSource used to send mail
    permission java.util.PropertyPermission "java.home", "read";
    permission java.util.PropertyPermission "java.naming.*", "read";
    permission java.util.PropertyPermission "javax.sql.*", "read";
    // OS Specific properties to allow read access
    permission java.util.PropertyPermission "os.name", "read";
    permission java.util.PropertyPermission "os.version", "read";
    permission java.util.PropertyPermission "os.arch", "read";
    permission java.util.PropertyPermission "file.separator", "read";
    permission java.util.PropertyPermission "path.separator", "read";
    permission java.util.PropertyPermission "line.separator", "read";
    // JVM properties to allow read access
    permission java.util.PropertyPermission "java.version", "read";
    permission java.util.PropertyPermission "java.vendor", "read";
    permission java.util.PropertyPermission "java.vendor.url", "read";
    permission java.util.PropertyPermission "java.class.version", "read";
    permission java.util.PropertyPermission "java.specification.version", "read";
    permission java.util.PropertyPermission "java.specification.vendor", "read";
    permission java.util.PropertyPermission "java.specification.name", "read";
    permission java.util.PropertyPermission "java.vm.specification.version", "read";
    permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
    permission java.util.PropertyPermission "java.vm.specification.name", "read";
    permission java.util.PropertyPermission "java.vm.version", "read";
    permission java.util.PropertyPermission "java.vm.vendor", "read";
    permission java.util.PropertyPermission "java.vm.name", "read";
    // Required for OpenJMX
    permission java.lang.RuntimePermission "getAttribute";
    // Allow read of JAXP compliant XML parser debug
    permission java.util.PropertyPermission "jaxp.debug", "read";
}
```

```

    // Precompiled JSPs need access to this package.
    permission java.lang.RuntimePermission
"accessClassInPackage.org.apache.jasper.runtime";
    permission java.lang.RuntimePermission
"accessClassInPackage.org.apache.jasper.runtime.*";
};
// You can assign additional permissions to particular web applications by
// adding additional "grant" entries here, based on the code base for that
// application, /WEB-INF/classes/, or /WEB-INF/lib/ jar files.
//
// Different permissions can be granted to JSP pages, classes loaded from
// the /WEB-INF/classes/ directory, all jar files in the /WEB-INF/lib/
// directory, or even to individual jar files in the /WEB-INF/lib/ directory.
//
// For instance, assume that the standard "examples" application
// included a JDBC driver that needed to establish a network connection to the
// corresponding database and used the scrape taglib to get the weather from
// the NOAA web server. You might create a "grant" entries like this:
//

// The permissions granted to the context root directory apply to JSP pages.
// grant codeBase "file:${catalina.home}/webapps/examples/-" {
//     permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
//     permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
//
// The permissions granted to the context WEB-INF/classes directory
// grant codeBase "file:${catalina.home}/webapps/examples/WEB-INF/classes/-" {
// };
//
// The permission granted to your JDBC driver
// grant codeBase "jar:file:${catalina.home}/webapps/examples/WEB-INF/lib/driver.jar!/-" {
//     permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
// };
// The permission granted to the scrape taglib
// grant codeBase "jar:file:${catalina.home}/webapps/examples/WEB-INF/lib/scrape.jar!/-" {
//     permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
grant codeBase "file:${catalina.home}/webapps/de/-" {permission
java.security.AllPermission;
};

```

Installing Directory Editor Using Tomcat Manager

This section explains how to install Directory Editor using Tomcat Manager:

NOTE You must have a manager role in Tomcat to use this installation method. To verify your status, check the following file:
file `<Tomcat_base_directory>/conf/tomcat-users.xml`

Use the following steps to install Directory Editor using the Tomcat Manager:

1. Navigate to the Tomcat `bin` directory and use one of the following methods to start Tomcat:
 - o **On Unix:** From the command line, invoke `startup.sh`.
 - o **On Windows:** Double-click `startup.bat`.
2. Open your browser and type `localhost:<port_number>/manager/html` into the URL field.

NOTE If you are running your web browser on a host other than the application server you may have to change or adjust the host name in the URL.

3. When you are prompted for your Tomcat Manager Application user name and password, enter the information and then press OK.
The Tomcat Manager page is displayed.
4. Scroll down until you locate the Deploy section called “War file to deploy.”
5. Click Browse to locate the Directory Editor `de.war` file.
6. Select the `de.war` file and then click Deploy.

Installing Directory Editor Manually

Use the following steps to install Directory Editor manually:

1. Copy the Directory Editor `de.war` file from its current location into the Tomcat `webapps` directory. For example,

`C:\Tomcat\jakarta-tomcat-5.0.28\webapps\`

2. Navigate to the Tomcat `bin` directory and use one of the following methods to stop Tomcat:

- o **On Unix:** From the command line, invoke `shutdown.sh`.
- o **On Windows:** Double-click `shutdown.bat`.

3. From the same directory, use one of the following methods to restart Tomcat:

- o **On Unix:** From the command line, invoke `startup.sh`.
- o **On Windows:** Double-click `startup.bat`.

The Tomcat application will automatically detect the `de.war` file, and will install the Directory Editor program.

NOTE Be sure to wait for Tomcat to finish initializing before proceeding to the next step.

4. Open your browser and type `localhost:<port_number>/de/` into the URL field.

The Directory Editor Startup Properties page is displayed. Continue to “Step 4: Specify the Startup Properties” on page 28 for instructions.

Installing Directory Editor for WebLogic

Use the procedures described in this section to install Directory Editor for use with the BEA WebLogic application server. The information is organized as follows:

- “Configuring the WebLogic Software” on page 24
- “Installing Directory Editor” on page 24
- “Configuring a WebLogic 7x Server” on page 25
- “Configuring a WebLogic 8.1 SP1Server” on page 26

Configuring the WebLogic Software

If necessary, install WebLogic (using that product’s installation instructions) and select the domain that will be referenced when you install Directory Editor.

Installing Directory Editor

Use the following steps to install Directory Editor:

1. Copy the Directory Editor `de.war` file from its current location into the folder where you want to install Directory Editor.

NOTE The WebLogic Web application home directory is:

- **For Version 7x:** `<ServerHome>/user_projects/<DomainName>/applications`
 - **For Version 8.1 SP1:** `<ServerHome>/user_projects/domains/<DomainName>/applications`
-

2. The Application Home panel will display the location where Directory Editor will be installed. Click Next to begin installation.
3. Navigate to the WebLogic `bin` directory and use one of the following methods to stop WebLogic:
 - **On Unix:** From the command line, invoke `shutdown.sh`.
 - **On Windows:** Double-click `shutdown.bat`.
4. From the same directory, restart WebLogic using one of the following methods:
 - **On Unix:** From the command line, invoke `startup.sh`.
 - **On Windows:** Double-click `startup.bat`.

The WebLogic application will automatically detect the `de.war` file, and will install the Directory Editor program.

NOTE Be sure to wait for WebLogic to finish initializing before proceeding to the next step.

Next you must configure the WebLogic server for use with Directory Editor. Continue to one of the following sections for instructions:

- “Configuring a WebLogic 7x Server” on page 25
- “Configuring a WebLogic 8.1 SP1Server” on page 26

Configuring a WebLogic 7x Server

Use these steps to configure a WebLogic 7x server:

1. Start the WebLogic server.
2. Start the BEA WebLogic Administration Console.
3. In the left panel, expand Deployments and then click Web Applications.
The console displays the Web Applications panel.
4. Click Configure a new Web Application.
5. Using the links, locate the `de` folder and select it.
6. Specify the target server. Select a server from the Available Servers list, move it to the Target Servers area, and then click Configure and Deploy.
7. Click Deploy to deploy Directory Editor.
8. Open your browser and type `localhost:<port_number>/de` into the URL field. (The port number will vary.)

NOTE If you are running your web browser on a host other than the application server you may have to change or adjust the host name in the URL.

The Directory Editor Startup Properties page is displayed. Continue to “Step 4: Specify the Startup Properties” on page 28 for instructions.

Configuring a WebLogic 8.1 SP1 Server

Use these steps to configure a WebLogic 8.1 SP1 server:

1. Start the WebLogic server.
2. Start the BEA WebLogic Administration Console.
3. In the left panel, expand Deployments, and then choose Web Application Modules.

The console displays the Web Applications panel.
4. Click Deploy a new Web Application Module.
5. Using the links under applications, locate and select the `de` folder where you put the `de.war` file.
6. Click Target Module.
7. Review the Targets, Accessibility and Identity configuration, and make any necessary changes.
8. Click Deploy to deploy Directory Editor.
9. Open your browser and type `localhost:<port_number>/de` into the URL field. (The port number will vary.)

NOTE If you are running your web browser on a host other than the application server you may have to change or adjust the host name in the URL.

The Directory Editor Startup Properties page is displayed. Continue to “Step 4: Specify the Startup Properties” on page 28 for instructions.

Installing Directory Editor for WebSphere 5.1

Use the following steps to install Directory Editor for use with the IBM WebSphere 5.1 (or later) application server.

1. Start the application server.
2. Start the WebSphere administration console, and then select Applications > Install New Application.
3. Add the `de.war` file name in the Path:Local Path field.
4. Add the path to the Context Root for the Directory Editor installation (for example, `/de`), and then click Next.

5. Select the Generate Default Bindings option (using the default selections for Override and Virtual Host), and then click Next.
6. Install a new applications page. If you do not want to install the application in WebSphere's default location, enter the path to a different location into the Directory to Install Application field. For example:

```
c:\Program Files\WebSphere\AppServer\installedApps\Hostname
```
7. Be sure the Distribute Application and Use Binary Configuration options are selected.
8. Be sure the Create Mbeans for Resources and Deploy EJBs options are *not* selected.
9. Enter the name of the application in the Application Name field (the default is *de*).
10. Selecting the Enable class reloading option is optional. Click Next.
11. To prepare for the new application's installation, make sure the panel displays a line for the current release of Directory Editor, and that it maps to the appropriate virtual host. Click Next.
12. Be sure the panel displays a line for the current release of Directory Editor, and that it maps to the appropriate server. Click Next and then click Finish.
13. Click Save to Master Configuration to save the configuration.
14. Click Save, and then wait for the page to clear.
15. Select Applications >Enterprise Applications, and then click the application name (the name you specified in the Application Name field).
16. Be sure the Use Metadata From Binaries option is selected.
17. Select `PARENT_LAST` in the Classloader Mode field.
18. Select Application in the WAR Classloader Policy field.
19. Click Apply, and then click OK.
20. From the menu bar, click Save.
21. Click Save to save the changes to the Master Configuration.
22. Stop and restart the application server.

23. Open your browser and type `localhost:<port_number>/de` into the URL field. (The port number will vary.)

-
- NOTE**
- With some platforms, there is a performance impact if you use the JCE provided with that platform. If you experience a long start-up time, see page 180 in Chapter 10, “Error Logging and Troubleshooting”.
 - If you are running your web browser on a host other than the application server you may have to change or adjust the host name in the URL.
-

The Directory Editor Startup Properties page is displayed. Continue to “Step 4: Specify the Startup Properties” on page 28 for instructions.

Step 4: Specify the Startup Properties

The first time you open Directory Editor, a Startup Properties page is displayed, similar to the following:

Figure 2-4 Startup Properties Page

Startup Properties

These settings are used by this application during initialization.

* Indicates required field

Configuration Directory Server

* Hosts (comma-separated):

* Port: Secure Port

* Bind DN:

* Password:

* Configuration Suffix:

Specify the host(s), port, and credentials and press Refresh.

Startup Options

Allow users to see this page during startup.

Allow users to log in anonymously.

Show user detailed message for failed log in attempts.

You will be prompted to specify these Startup properties and Managed Directory properties (described in the next section).

NOTE Directory Editor may automatically complete some of the properties fields, but you can change the information if necessary.

Use the following information to complete the Startup Properties page:

1. Specify the following Configuration Directory Server parameters:
 - **Host:** Enter the name of the host(s) where your configuration Directory Server is located.

The Host field is a multi-valued field that accepts a comma-delimited list of host names.

If you specify more than one host name, you must configure the Directory Servers on each host to replicate master-to-master. Directory Editor will load-balance and provide failover between the configured hosts.

In addition, these hosts *must* be running Directory Server on the *same port* with the *same security setting*. For example, port 389 with nonsecure connections.

NOTE The Directory Editor *configuration directory* is a directory where you store the product's configuration information. This directory server does not have to be a configuration directory as defined by the *Sun ONE Server Console 5.x Server Management Guide*.

- **Port:** Enter the port number on which the Directory Server is listening.

CAUTION If you use SSL to connect to a server on a port that is not using SSL, the connection will hang — which is a characteristic of the SSL protocol.

Similarly, using a plain socket to connect to a server's SSL socket, will also cause the connection to hang.

- **Bind DN:** Enter the bind distinguished name used to authenticate to Directory Server in the bind request.

- **Password:** Enter the password you use to access the configuration directory.

NOTE After completing the Configuration Directory Server Properties section, click the Refresh button and Directory Editor will automatically display the naming contexts available for storing the configuration.

- **Configuration Suffix:** Specify the base suffix of the naming context where the Directory Editor configuration is stored.

2. Specify the Startup Options, as follows:

- **Allow users to see this page during startup:** Enable or disable the checkbox to control whether the Startup Properties page is displayed to end users.

NOTE The Startup Properties page is displayed if Directory Editor cannot connect to the configuration directory when the user first starts the application.

For security purposes, disable this checkbox after the initial Directory Editor installation. After disabling this checkbox, you can edit the configuration directory settings by selecting Configuration > Startup Page or by editing the `startup.properties` file in the `WEB-INF` directory.

- **Allow users to log in anonymously:** Enable or disable the checkbox to control whether your users can log in to Directory Editor anonymously.

If users log in anonymously, they can access Directory Editor's Home, Browse, and Search pages only. They will not have access to the Create or Configure pages.

NOTE After a user logs in, Directory Editor will not allow them to delete, disable, or rename the object (dn) they used to login. If they try to change the object an error message will result.

- **Show user detailed message for failed log in attempts:** Enable or disable the checkbox to control whether the end-user will see more-detailed failed log-in messages.

For example, if the user enters an invalid password

- The following message displays if this option is disabled:

Authentication Failed: Invalid Credentials

- The following message displays if this option is enabled:

Authentication Failed: Invalid Password

NOTE **For Advanced Users Only:**

The following properties are configurable, but they are not represented in the Directory Editor user interface:

- **datastore.objectClass:** Specify the LDAP object class that Directory Editor can use to store configuration objects in the directory.
- **datastore.dmlIdAttribute:** Specify an attribute from the specified object class that Directory Editor can use to store configuration object IDs.
- **datastore.xmlObjectAttribute:** Specify an attribute from the specified object class that Directory Editor can use to store configuration content in XML format.
- **datastore.location:** Specify the configuration location when it is prepended with the `datastore.configSuffix`.

You can configure these properties in the `startup.properties` file located in `<application_home>\WEB-INF`; however, changing these attributes, may cause issues with the program and can make support more difficult.

3. When you are finished with this page, click the Save and Continue button to save the information.

NOTE Your application server must allow write access to Directory Editor's WEB-INF directory to persist startup.properties.

If your application server does not allow write access, you will be allowed to continue, but Directory Editor will remain in *non-production* mode.

NOTE After this initial configuration of the Startup Properties page, you can edit any of the configuration parameters by selecting Configuration > Startup.

A Managed Directory page displays (similar to the following figure).

Figure 2-5 Managed Directory Page

Instructions for completing this page are provided in the next section.

Step 5: Specify the Managed Directory Properties

Use the following information to complete the Managed Directory page (Figure 2-5):

1. Specify the following parameters:

- **Host:** Enter the name of the host(s) where your managed directory is located.

To specify more than one Host, click the Add button located next to the Host field . (To remove a host, click the Rem button.)

NOTE If you specify more than one host name, you must configure the Directory Servers on each host to replicate master-to-master. Directory Editor will load-balance and provide failover between the configured hosts.

In addition, these hosts *must* be running Directory Server on the *same port* with the *same security setting*. For example, port 389 with nonsecure connections.

- **Port:** Enter the port number on which the managed directory is listening, and then enable or disable the Secure Port checkbox to control whether this directory must communicate using a secure connection.

CAUTION If you use SSL to connect to a server on a port that is not using SSL, the connection will hang — which is a characteristic of the SSL protocol.

Similarly, using a plain socket to connect to a server's SSL socket, will also cause the connection to hang.

- **Base Context:** Click the Refresh button to populate the drop-down menu.

NOTE You must fill out the host and port fields before pressing Refresh.

Use the menu to select a base context to be managed by Directory Editor. (For example, `ou=People,dc=example,dc=com`)

NOTE Be sure to set the base context high enough in the tree to ensure that you have access to all the information you need.

- **Manager Principals:** Enter the name or distinguished name (DN) of one or more LDAP groups under the base context whose members are considered Directory Editor directory managers (administrators). These directory managers will have access rights to all Directory Editor functionality. (For more information, see “Configuring Directory Editor” on page 141.)

For additional manager principals, click the Add button. To remove principals, click the Rem button.



NOTE After completing the initial configuration of Directory Editor, you will be able to browse the directory for manager principals to add to the Manager Principals list.

(For more information, see “Editing the Managed Directory Properties” on page 157.)

- **User search authentication:** Enter the method by which Directory Editor will search the directory for authenticating users.
 - **Anonymously:** Enable this button if your user objects are visible to anonymous search queries. (The Bind DN and Password text fields will become inactive and you cannot type in those fields.)
 - **Simple Bind:** Enable this button and then specify a Bind DN and Password if your user objects are not visible to anonymous search queries.

- **Bind DN:** Enter the bind distinguished name used to authenticate to the managed directory in the bind request (*not required for anonymous user search authentication*). This option enables Directory Editor to search for users during the login process only.
- **Password:** Enter the password you use to access your managed directory to search for user object DNs (*not required for anonymous user search authentication*).
- **Naming Attributes:** Enter the attributes used in the directory tree. For additional attributes, click the Add button. To remove attributes, click the Rem button.

When a user tries to log in, Directory Editor uses the Account ID field on the Log In form to search for a user object that matches one of these naming attributes exactly.

2. When you are finished, click Save to save the information and to open the Directory Editor Log In page.

NOTE After this initial configuration of the Managed Directory page, you can edit any of the configuration parameters by selecting Configuration > Managed Directory.

What's Next?

Continue to Chapter 3, “Getting Started” to log-in and begin customizing applications with Directory Editor.

What's Next?

Getting Started

This chapter provides an overview of Directory Editor's graphical user interface and major features. The information in this chapter is organized as follows:

- “Product Features” on page 37
- “Logging In” on page 38
- “Changing Your LDAP Password” on page 39
- “Navigating Directory Editor” on page 40
- “Browsing Directories” on page 42
- “Using the Browse Button” on page 44
- “Using Online Help” on page 46

Product Features

Directory Editor is J2EE-based web application that enables you to quickly and easily manage directory data. Directory Editor:

- Provides an easy-to-use, directory editing tool that is appropriate for administrators and end-users
- Provides ability to create/edit users, groups, containers, and more
- Supports a many concurrent users and large directory installations
- Supports extensive customizations, branding, and embedding
- Supports private data transmissions with the client browser and the directory
- Provides built-in authorization to limit visible menus to users roles

Logging In

When the Directory Editor Log In page is displayed (see Figure 3-1), enter your credentials (user name and password) for full access.

Figure 3-1 Directory Editor Log In Page



Sun Java™ System Directory Editor

Account Id:

Password:

Log In Anonymously

Copyright ©2004 Sun Microsystems, Inc. All rights reserved. U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

NOTE

- If you enable the Log In Anonymously checkbox, you will have access to Directory Editor's Home, Browse, and Search pages only. You cannot access the Create or Configure pages.
 - After initial configuration, the log in screen (Figure 3-1) will display every time you open Directory Editor.
 - After a user logs in, Directory Editor will not allow them to delete, disable, or rename the object (dn) they used to login. If they try to change the object an error message will result.
-

When you log in successfully, the Directory Editor Home page is displayed. (See "Navigating Directory Editor" on page 40 for an overview of this, and other, Directory Editor pages.)

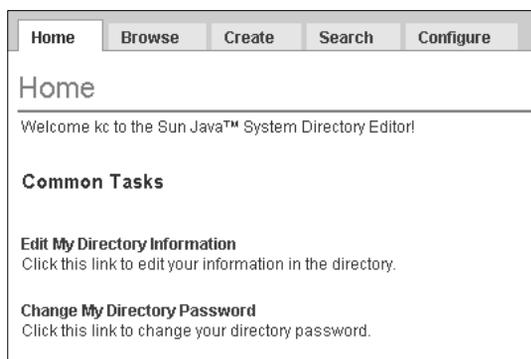
Changing Your LDAP Password

NOTE If your password is about to expire, Directory Editor provides a warning message and allows you to change your password. However; if your password has already expired, you cannot log in until the system administrator resets your password.

To change your LDAP password, use the following steps:

1. From the Home page, click the Change My Directory Password link.

Figure 3-2 Change My Directory Password Link



2. When the Edit Password page is displayed (Figure 3-3), type a new password into the Password text box and the Confirm Password text box.

Figure 3-3 Edit Password Page

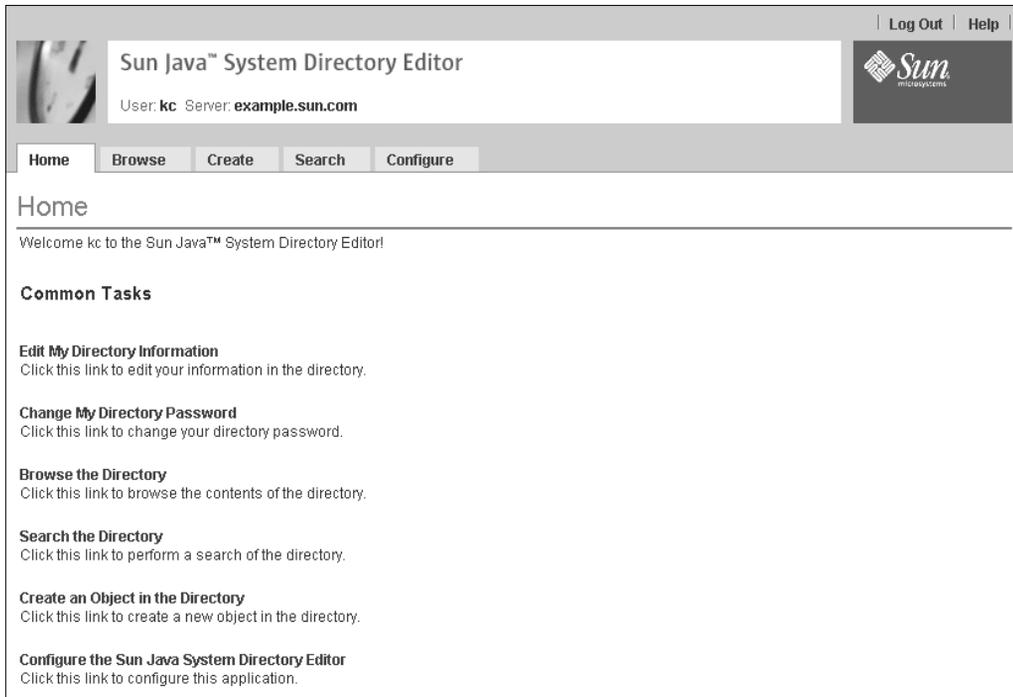
The screenshot shows the 'Edit Password' page. At the top right, there are 'Save' and 'Cancel' buttons. The main content area contains the instruction 'Type a new LDAP password.' followed by two text input fields: 'Password:' and 'Confirm Password:'.

3. When you are finished, click Save to save the new password and return to the Home page (or click Cancel to return to the Home page without saving the password).

Navigating Directory Editor

After logging into Directory Editor as an administrator, the Home page displays as shown in Figure 3-4:

Figure 3-4 Directory Editor Home Page



NOTE If your Home page looks different than the one shown in Figure 3-4, you may have specified credentials for an account that is not a member of the Manager Group you specified in “Step 5: Specify the Managed Directory Properties” on page 33.

The Home page contains a list of tasks commonly performed with Directory Editor. Click any of the links on this page to perform the task described, as follows:

- **Edit My Directory Information:** Click this link to open the Edit an Object page, where you can edit your information only. (See “Editing Objects” on page 86.)
- **Change My Directory Password:** Click this link to open the Edit Password page, where you can change your LDAP password. (See “Changing Your LDAP Password” on page 39.)
- **Browse the Directory:** Click this link or select the Browse tab to open the Browse page where you can view the contents of the directory in tree format. (See “Browsing Directories” on page 42.)
- **Search the Directory:** Click this link or select the Search tab to open the Search page, where you can to locate objects in the directory. (See Chapter 6, “Searching Directories.”)

You can execute three different types of searches from the Search page:

- **Basic Search** (*default*): Search for a user, a group, or an organizational unit.
- **Advanced Search:** Specify conditions that can be combined with *and* operations to search for objects.
- **Filtered Search:** Use a free-form query string to search for objects.
- **Create an Object in the Directory:** Click this link or select the Create tab to open the Create page where you can create user, group, organizational unit, domain, and locality objects. (See Chapter 4, “Creating and Editing Objects.”)
- **Configure the Sun Java™ System Directory Editor** (*available to administrative users only*): Click this link or select the Configure tab to open the Configure page where you can configure the Directory Editor application. (See Chapter 7, “Configuring Directory Editor.”)

Configuring the application includes:

- Customizing forms
- Configuring the Startup and Managed directories
- Configuring searches
- Configuring naming attributes
- Specifying roles
- Backing up and restoring configurations

The following options (located in the upper-right corner of the application window) are available from all Directory Editor web pages:

- **Log Out:** Click the link to log out of the Directory Editor application. (The Directory Editor log in page is displayed immediately.)
- **Help:** Click the link to open the online Help when you need assistance with the interface or a procedure. (Instructions for using the Help tool are provided in “Using Online Help” on page 46.)

Browsing Directories

Select the Browse tab to browse the contents of the directory.

Initially, only the top-level directory objects (based on object class) are displayed in tree view, as follows:

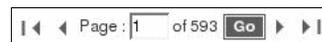
-  OU, O
-  Domains
-  Groups
-  Person
-  Role
-  All other

NOTE You specify the directory context using the Managed Directory parameters stored in the configuration directory.

Figure 3-5 Browse Tab

Click the node key  to view the contents of that folder.

If there are more than 100 objects in the folder, the objects will be displayed on several pages, and the following navigation tool will display.



Use this navigation tool to view the different pages as follows:

- Enter a page number into the text box and click the Go button to see a particular page in the set.
- Click  to go forward one page or  to go back one page.
- Click  to go to the first page in the set or click  to go to the last page.

In addition, Directory Editor provides a link to the Search page. Click the link if you want to search for a particular object.

NOTE

- If you click the folder icon or directory name, the Edit page is displayed so you can edit that object.

For example, if you select `cn=Directory Administrators` you can edit the object name, the availability of people in the group, add or remove members, and view statistics. Instructions for editing objects are provided in Chapter 4, “Creating and Editing Objects.”

- You can change the default settings for maximum number of lines to display per page and page size (as well as other settings) by selecting `Configure > Forms` and then click the `Default Browse Form's Customize` button. When the next page displays, click the `ldaptree` node. You can edit any of the parameters listed.

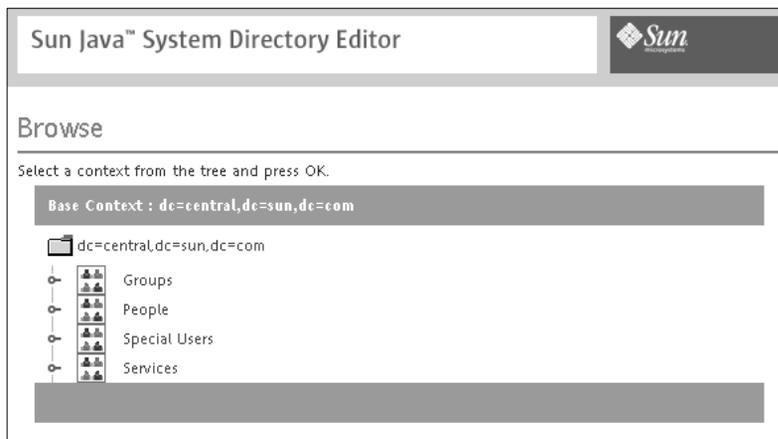
For more information, see Chapter 5, “Working With Forms.”

Using the Browse Button

Many Directory Editor pages provide a Browse button to help you locate information you must provide to complete a task.

For example, to create a new object, you must provide a parent entry (the name of the context) where you are going to add the object. You could use the following steps to locate the parent:

1. Click the Browse button to open a Browse dialog box (Figure 3-6):

Figure 3-6 Example Browse Dialog Box

2. In this dialog box, navigate within the tree to locate the appropriate parent entry (or context name).
3. Click the entry to select it.

For example, you might want to add a new user entry under the People node.

4. When you have made your selection, click OK to save your selection and return to the Create page.

Directory Editor then enters the context information into the Parent Entry text box.

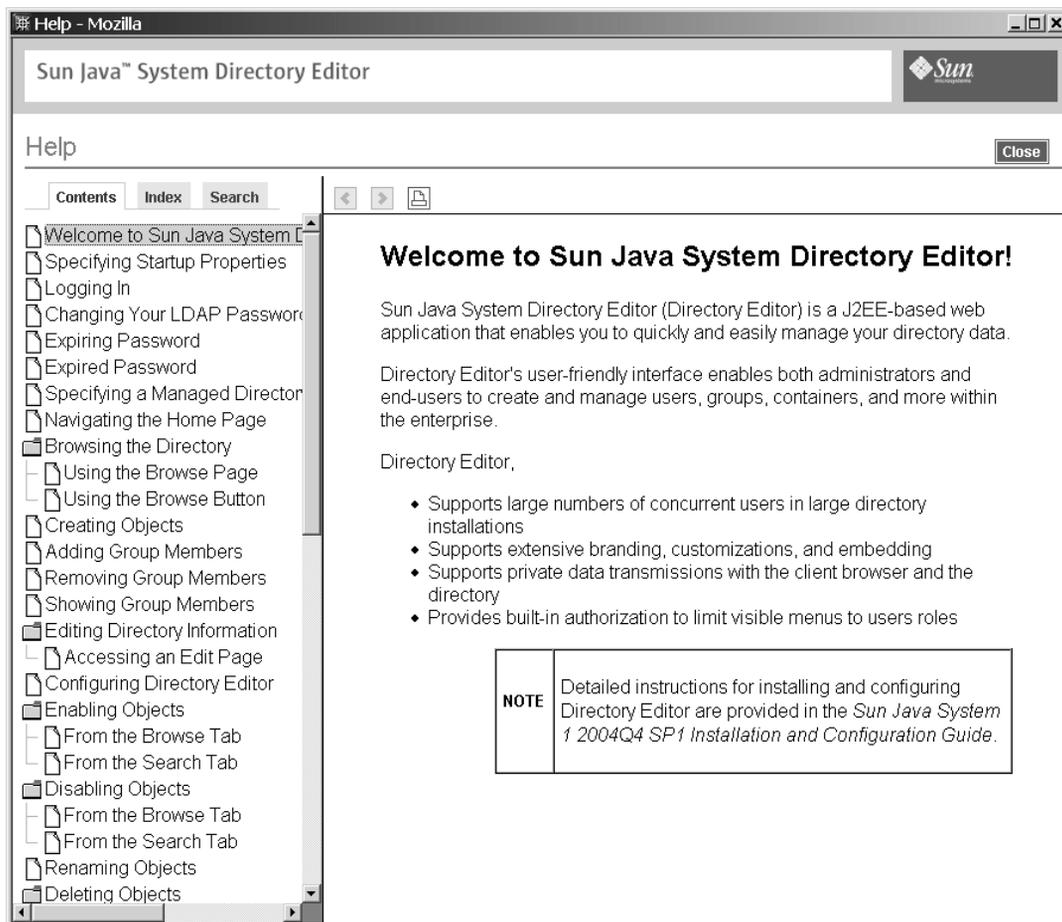
Figure 3-7 Completed Parent Entry Text Box

Using Online Help

Directory Editor provides an online help tool that you can use to quickly and easily locate information about the product.

To access the online help, click the Help link located in the upper right corner of the Directory Editor browser window and the Help window is displayed as follows:

Figure 3-8 Online Help Window



Use the following features to navigate in the Help window and to locate information about Directory Editor:

- **Contents tab:** Lists all available help topics. Click a topic name to display the help content in the information pane (located on the right side of the Help window).
- **Index tab:** Contains an index of terms and topics. Click an index entry to display the help content in the information pane.
- **Search tab:** Enter a text string into the text box and press the Return (Enter) key to search for help content associated with that string.
- **Back and Forward buttons:** Click the Back button  to display to the previous help topic in the information pane or click the Forward button  to display to the next help topic.
- **Print button:** Click the Print button  to print the content displayed in the right pane. A standard Print dialog will display so you can specify print settings and select a printer.

When you are finished with the Help window, click the Close button to close the Help window.

Creating and Editing Objects

You can use the Create tab to create new users, groups, organizational units, domains, or localities. This chapter explains how to create each object, and the information is organized as follows:

- “Creating Objects” on page 49
- “Editing Objects” on page 86

Creating Objects

This section explains how to access the Create page and create objects using Directory Editor. The information is organized as follows:

- “Accessing the Create Page” on page 50
- “Creating Users” on page 50
- “Creating Groups” on page 64
- “Creating Organizational Units” on page 75
- “Creating Domains” on page 80
- “Creating Localities” on page 83

NOTE If Directory Server does not have a schema entry for the object you want to create, you must do the following:

To create an object (or the form for an object), you must add the required object classes to the schema. Directory Editor reads the schema entries to build the form fields.

Accessing the Create Page

You can access the Create Page using one of the following methods:

- Select the Create Tab
- Select the Browse tab, click a node in the tree, and then click the Create button.
- Select the Search tab, execute a search from any of the Search tabs (Basic, Advanced, or Filtered), and click the Create button located just above the Results table. (See Chapter 6, “Searching Directories.”)

Each method opens the Create page shown in Figure 4-1.

Figure 4-1 Create Page

Creating Users

Use the following steps to create a new user:

1. Select the Create tab.
2. On the Create page (Figure 4-1), enable the User radio button.
3. Indicate where to add the new user by typing the parent’s context information into the Parent Entry text field (or click Browse to locate the information). For example, to create the new user under People, you might enter:

```
ou=People,dc=example,dc=com
```
4. When you are done, click Continue (or press Enter/Return on your keyboard).

NOTE Clicking the Cancel button clears the Parent Entry text box.

A new Create page displays and defaults to the Quick Create tab (Figure 4-2).

Figure 4-2 Quick Create Tab

- The Quick Create tab contains all of the required attributes you must provide to create the object. For example, to create an User object, you must enter a user name, first name, last name, and at least one common name for that user.
 - The other tabs on this page enable you to specify additional, *but optional*, information about the new user.
5. You can proceed in one of two ways:
- Enter the required information in the text boxes provided on the Quick Create tab to create the new user, and then click Save to apply your changes.
 - After providing the required information, you can select one or more of the other tabs to provide additional information about the new user.

The following sections provide instructions for completing each tab on the Create page:

- “Completing the Common Attributes Tab” on page 52
 - “Completing the Group Membership Tab” on page 54
 - “Completing the Location Info Tab” on page 55
 - “Completing the Misc Attributes Tab” on page 56
 - “Completing the Extensions Tab” on page 57
6. When you are finished with the Create page (being sure to provide all *required* information), click Save to add the new user to the directory.

Completing the Common Attributes Tab

To provide general information about the new user, use the following steps:

1. Select the Common Attributes tab and complete one or more of the following text boxes:

-
- NOTE**
- The program will automatically insert the user name, first name, last name, and common names if you entered them on the Quick Create tab (page 51).
 - Use the Add and Rem buttons located to the right of some attributes on this page to enter multiple values for those attributes or to remove values. For example, you might want to enter multiple telephone numbers (work, mobile, and home).
-

- **User Name:** Enter the user name (uid).
- **First Name** (*required*): Enter the user's first name.
- **Last Name** (*required*): Enter the user's last name.
- **Common Name:** Enter at least one common name if it is different from the user's first name.
For example, Kimberly Jones may prefer being called "Kim Jones."
- **Password:** Type a password for the new user.
- **Confirm Password:** Re-type the password to confirm it.
- **Email:** Enter the user's email address.
- **Telephone Number:** Enter the user's telephone number.
- **Mobile Phone:** Enter one or more mobile telephone numbers.
- **Pager:** Enter one or more pager numbers.
- **Facsimile Telephone Number:** Enter one or more fax numbers.
- **Department #:** Enter the user's department number.
- **Employee #:** Enter the user's employee number.
- **Employee Type:** Enter information about the user's employment status (such as salaried, contract, part-time, and so forth).

For example, if you are creating a new user named *Mike Miller*, your page might look something like the one in the following figure.

Figure 4-3 Mike Miller's Common Attributes

The screenshot shows a web form titled "User - Create" with a "Save" and "Cancel" button in the top right. Below the title is a tabbed interface with six tabs: "Quick Create", "Common Attributes", "Group Membership", "Location Info", "Misc Attributes", and "Extensions". The "Common Attributes" tab is selected. The form contains the following fields:

- * User Name:
- * First Name:
- * Last Name:
- * Common Name: ⊕ Add
- Password:
- Confirm Password:
- Email: ⊕ Add
- Telephone Number: ⊕ Add
- Mobile Phone: ⊕ Add
- Pager: ⊕ Add
- Facsimile Telephone Number: ⊕ Add
- Department #:
- Employee #:
- Employee Type:

* Indicates required field

2. When you are finished with this tab, you can
 - Select another tab to provide additional information.
 - Click Save to save the new information to the directory.
 - Click Cancel to clear the Common Attributes text boxes.

Completing the Group Membership Tab

To assign a new user to specific groups, use the following steps:

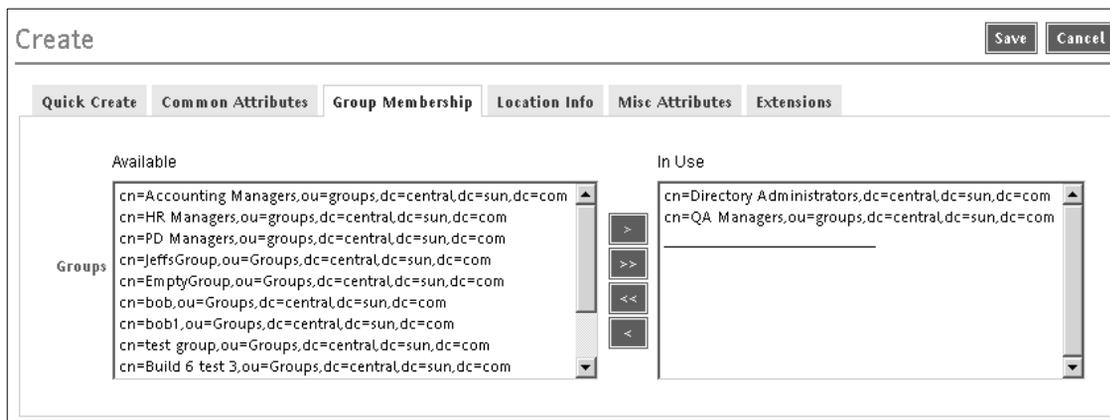
1. Select the Group Membership tab.
2. Use the Groups selection tool as follows:

TIP Press the Shift key on your keyboard, and then click on group names to select multiple names in either list.

- To add to one or more groups to the In Use list, select the group names in the Available list and click the  button.
- To move all available groups to the In Use list, click the  button.
- To move all groups in the In Use list back to the Available list, click the  button.
- To move one or more groups back to the Available list, select the group names in the In Use list and click the  button.

For example, to assign Mike Miller to the Directory Administrators group and the QA Managers group, select those two groups from the Available list and add them to the In Use List, as shown in the following figure:

Figure 4-4 Example Group Membership Tab



3. When you are finished with this tab, you can
 - Select another tab to provide additional information.
 - Click Save to save the new information to the directory.
 - Click Cancel to clear your changes and return to the initial Create page.

Completing the Location Info Tab

To specify the new user's physical location in the company, use the following steps:

1. Select the Location Info tab.
2. Type the following information in the text boxes provided:
 - **Postal Address:** Enter the user's mailing address.
 - **Mail stop:** Enter the user's company mail stop.
 - **Postal Code:** Enter the user's mailing address postal code (zip code).

For example, you might enter the following information for Mike Miller:

Figure 4-5 Example Location Information

Postal Address:	111 A Street, Suite 1, Anywhere, CA
Mailstop:	B1111
Postal Code:	11111-0000

3. When you are finished with this tab, you can
 - Select another tab to provide additional information.
 - Click Save to save the new information to the directory.
 - Click Cancel to clear the text boxes.

Completing the Misc Attributes Tab

To provide additional information about the new user, use the following steps:

1. Select the Misc Attributes tab.
2. Type the following information in the text boxes provided:
 - **Photo URL:** Enter a URL location where the user's photograph can be viewed.
 - **Home Phone:** Enter one or more home telephone numbers.
 - **Home Address:** Enter the user's home address.
 - **Auto License:** Enter the user's automobile license number.

For example, you might enter the following information for Mike Miller:

Figure 4-6 Example Misc Attributes Tab

NOTE If the new user has multiple home phones or cars, click the Add button to create new, blank text boxes so you can enter the additional numbers. Repeat as many times as necessary.

If you add new fields, Rem buttons will also display next to each field (as shown in the following figure). Click the Rem button to remove extra fields.

3. When you are finished with this tab, you can
 - Select another tab to provide additional information.
 - Click Save to save the new information to the directory.
 - Click Cancel to clear the text boxes.

Completing the Extensions Tab

Use the Extensions tab to specify additional attributes to be made available for the new user.

NOTE The `inetOrgPerson`, `organizationalPerson`, `person`, and `top` extensions are in use by default.

To specify extensions, use the following steps:

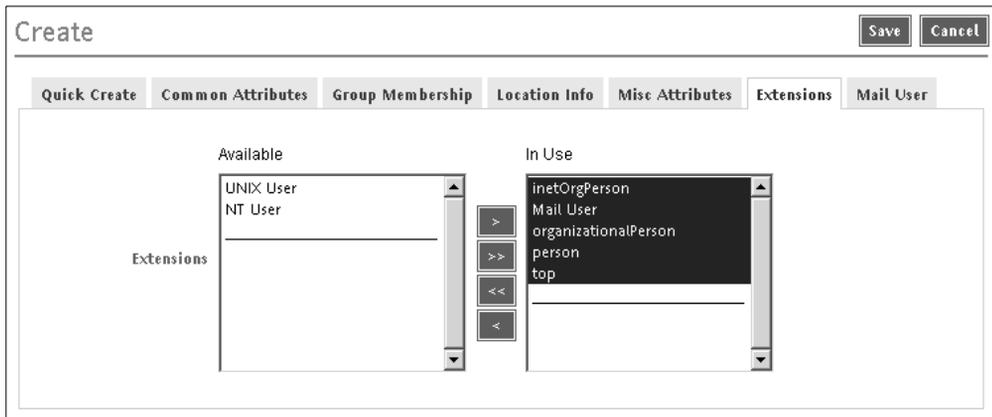
1. Select the Extensions tab.
2. Use the Extensions selection tool as follows:

TIP Press the Shift key on your keyboard, and then click on group names to select multiple names in either list.

- To add to one or more extensions to the In Use list, select the extension name(s) in the Available list and click the  button.
- To move all available extensions to the In Use list, click the  button.
- To move all extensions in the In Use list back to the Available list, click the  button.
- To move one or more extensions back to the Available list, select the extension name(s) in the In Use list and click the  button.

For example, you might want to select Mail User:

Figure 4-7 Example Extensions Tab



NOTE When you add Mail User, NT User, and/or UNIX User extensions to the In Use List, Directory Editor adds tabs to the Create page so you can provide additional information related to those extensions. See the following sections for instructions:

3. When you are finished with this tab, you can
 - o Select another tab to provide additional information.
 - o Click Save to save the new information to the directory.
 - o Click Cancel to clear the text boxes.

Completing the Mail User Tab

Use the Mail User tab to provide detailed information about the new user's Mail account, as follows:

-
- NOTE**
- The program will automatically insert `cn`, `mail`, `uid`, and `userPassword` values based on the information you entered on the Quick Create tab (page 4-35), and on the Common Attributes tab (page 4-36).
 - For detailed information about specifying the attributes on this page, consult the *Sun Java™ System Directory Server 5 2004Q2 Administration Reference*.
-

1. Select the Mail User tab.
2. Enter the following user information:

Table 4-1 Mail User Attributes

Attribute Name	Description
<code>cn</code>	User's common name
<code>mail</code>	User's primary email address
<code>mailAlternateAddress</code>	Alternate email addresses
<code>mailHost</code>	Fully qualified DNS hostname of the user's email host
<code>mailRoutingAddress</code>	Forwarding email address for the user
<code>mailAccessDomain</code>	One or more domains, server names, or IP address the user can use to log on via POP or IMAP
<code>mailAutoReplyMode</code>	Auto-reply mode (For example, <i>vacation</i>)
<code>mailAutoReplyText</code>	An automatic email reply message
<code>mailDeliveryOption</code>	How the user's email should be delivered
<code>mailForwardingAddress</code>	Forwarding email address
<code>mailMessageStore</code>	Absolute path to the email host directory that contains the message store where the user's mailbox resides
<code>mailProgramDeliveryInfo</code>	One or more commands to specify programmed mail delivery for the user
<code>mailQuota</code>	Value to indicate the maximum disk space (in bytes) available to the user's mailbox
<code>multiLineDescription</code>	Description of the user

Table 4-1 Mail User Attributes *(Continued)*

uid	User's user ID
userPassword	Temporary email password
userPassword Confirmation	Confirm password.

NOTE To add multiple email addresses, click the Add button located next to the mail text field. To remove an email address, click the Rem button.

For example, you might enter the following information for Mike Miller:

Figure 4-8 Example Mail User Tab

The screenshot shows a configuration window with several tabs: Quick Create, Common Attributes, Group Membership, Location Info, Misc Attributes, Extensions, and Mail User. The Mail User tab is active, displaying a list of attributes and their values for a user named Michael Miller. The attributes and their values are as follows:

- cn:** Michael Miller (with an Add button)
- mail:** mmiller@example.com (with an Add button)
- mailAlternateAddress:** mmm11111@newaddress.net
- mailHost:** lorre.example.com
- mailRoutingAddress:** (empty)
- mailAccessDomain:** example.com
- mailAutoReplyMode:** vacation
- mailAutoReplyText:** On vacation until Jan. 5, 2005.\$See you next year
- mailDeliveryOption:** mailbox
- mailForwardingAddress:** (empty)
- mailMessageStore:** (empty)
- mailProgramDeliveryInfo:** (empty)
- mailQuota:** 5000000
- multiLineDescription:** (empty)
- uid:** mmiller
- userPassword:** (masked with asterisks)
- userPassword Confirmation:** (masked with asterisks)

-
- NOTE**
- If the new user has multiple email addresses, click the Add button to create new blank fields so you can enter the additional addresses. Repeat as many times as necessary.
 - When you add new fields, Rem buttons are also displayed next to each field. Click the Rem button to remove extra fields.
-

3. When you are finished with this tab, you can
 - Select another tab to provide additional information.
 - Click Save to save the new user information to the directory.
 - Click Cancel to clear the text boxes.

Completing the Default NT User Tab

Use the Default NT User tab to define entries for the new user that describe the user's Windows NT account.

To specify these entries, use the following steps:

1. Select the Default NT User tab.
2. Enter the following user information:

-
- NOTE**
- NT User Id is the only required attribute on this tab.
 - For detailed information about specifying the attributes on this page, consult the *Sun Java™ System Directory Server 5 2004Q2 Administration Reference*.
-

Table 4-2 NT User Attributes

Attribute Name	Description
NT User ID	User's NT domain name
Comment	Description about user's NT account
User Workstation List	NT workstations user can log into within your NT domain
Home Drive	Drive letter assigned to user's home directory
Home Directory	User's NT home directory

Table 4-2 NT User Attributes *(Continued)*

Server	Server to which user's NT logon requests are sent
User Profile Path	Path to user's Windows NT profile
Logon Script	Path to user's NT login script

For example, you might enter the following information for Mike Miller:

Figure 4-9 Example NT User Tab

The screenshot shows a 'Create' dialog box with a 'Default NT User Form' tab selected. The form contains the following fields and values:

- * NT User ID:** workgroup:mmiller
- Comment:** QA manager for Bogart 1.0 project
- User Workstation List:** bacall
- Home Drive:** c:
- Home Directory:** c:\home\example\mmiller
- Server:** \\bacall
- User Profile Path:** (empty)
- Logon Script:** c:\home\example\mmiller\script.bat

A legend indicates that an asterisk (*) denotes a required field. The dialog box also features 'Save' and 'Cancel' buttons in the top right corner.

3. When you are finished with this tab, you can
 - o Select another tab to provide additional information.
 - o Click Save to save the new user information to the directory.
 - o Click Cancel to clear the text boxes.

Completing the UNIX User Tab

Use the UNIX User tab to provide detailed information about the new user's UNIX account, as follows:

-
- NOTE**
- The program automatically inserts values you entered on the Quick Create tab (page 4-35) and the temporary user password you entered on the Common Attributes tab (page 4-36).
 - For detailed information about specifying the attributes on this page, consult the *Sun Java™ System Directory Server 5 2004Q2 Administration Reference*.
-

Table 4-3 Unix User Attributes

Attribute Name	Description
cn	User's common name
uid	User's user ID
uidNumber	Value that uniquely identifies the user in your administrative domain
gidNumber	Value that uniquely identifies the user's group in your administrative domain
homeDirectory	Absolute path to user's home directory
userPassword	Temporary password for new user
userPassword Confirmation	Confirm password
loginShell	Path to user's login shell
gecos	GECOS field (or common name)
description	Information about user, such as work assignment

For example, you might enter the following information for Mike Miller:

Figure 4-10 Example UNIX User Tab

The screenshot shows a tabbed interface with the 'UNIX User' tab selected. The fields and their values are as follows:

Field	Value
* cn	Michael Miller
* uid	mmiller
* uidNumber	658
* gidNumber	650
* homeDirectory	/home/mmiller111
userPassword	*****
userPassword Confirmation	*****
loginShell	
gecos	
description	QA Manager for Bogart 1.0 project

* Indicates required field

Creating Groups

Use the following steps to create a new group object in the directory:

1. From the Create tab, enable the Group radio button.
2. Indicate where to add the new group by typing the parent's context information into the Parent Entry text field as shown in Figure 4-11 (or click Browse to locate the information).

For example, you might enter:

```
ou=Groups,dc=example,dc=com
```

Figure 4-11 Create Page

3. When you are finished, click Continue.

NOTE The Cancel button clears the Organization text box.

The Quick Create page is displayed (see Figure 4-12).

Figure 4-12 Quick Create Tab

-
- NOTE**
- By default, you will be creating a static group object (an object with the `groupOfNames` or `groupOfUniqueNames` object class). However, you can also create dynamic group objects (a group that determines membership at any given moment using a set of LDAP search filters). For instructions, see “Completing the Dynamic Group Tab” on page 72.
 - The Quick Create tab contains all of the attributes you must provide to create the selected object type. For example, to create a Group, you must enter a Static Group name.
 - The other tabs on this page enable you to specify additional, but optional, information about the object.
-

4. You can proceed in one of two ways:
 - Type a name in the Name text box provided on the Quick Create tab to create the new group, and then click Save.

Now, if you click the Groups node on the Browse page, the new group name is displayed in the tree.
 - After providing the required information, select one or more of the other tabs on the Create page to provide additional information about the new group.

The following sections provide instructions for completing each tab on the Create page:

- “Completing the Static Group Tab” on page 66
 - “Completing the Import/Export Tab” on page 71
 - “Completing the Dynamic Group Tab” on page 72
5. When you are finished entering the new information on the Create page (being sure to provide all *required* information), click Save to add the new group to the directory.

Completing the Static Group Tab

A static group is a mail group defined statically by enumerating each group member.

You can use the Static Group tab to perform the following tasks:

- Add members to a static group
- Remove members from a static group (see page 69)
- Show members in a static group (see page 71)

Adding Members to the Group

To perform any of these tasks, use the following steps:

1. Select the Static Group tab.

Figure 4-13 Static Group Tab

2. If you specified a group name on the Quick Create tab, Directory Editor automatically completes the Name text box.

If you have not yet provided a name (or want to change the current name) enter a name in the text box.

3. Click the Add Members button and a Search page is displayed.

Figure 4-14 Search For Members to Add to the Group Search Page

4. You can use the Basic Search, Advanced Search, or Filtered Search tabs to search the directory for members to add to this new group.

NOTE Detailed instructions for using these tabs are provided in Chapter 6, “Searching Directories.”

To illustrate the procedure using the Basic Search tab, use one of the following methods:

- **To search for a particular member:** Type the member's name into the Enter the name of a User, Group, or Organizational Unit text box, use the Search Type menu to select the object type, and then click Search.

For example, enter Jones in the text box and select User (or All). Directory Editor will return all members named Jones.

- **To search for all the members:** Leave the text box blank, use the Search Type menu to select the object type, and then click Search.

For example, if you do not know the name of the member to search for, leave the text box empty and select User (or All). Directory Editor will return all User members.

5. When the Results table displays, enable the checkbox(es) next to the member(s) you want added to the new group.
6. Click the Add Selected To Group button.

Directory Editor will add the selected member(s) to the new group and return you to the Static Group tab.

NOTE The Number of Members value will not change to reflect the number of members added to the group until you click Save to save your changes.

7. You can proceed in several ways:
 - If you are finished adding members to the group (and you have no other tasks to perform on this page), click Save to save your changes.

NOTE Directory Editor does not save the new members to the directory until you click the Save button on the Static Group page.

Directory Editor returns you to the Browse page so you can verify that the new group was added with the specified members.

- Click Add Members to continue adding members to the new group.
- Click Remove Members and continue to “Removing Members from the Group” on page 69 to remove members from the group.

- Click Show Members and continue to “Showing Members in the Group” on page 71 to view the existing members of the group.
- Click the Import/Export tab and continue to “Completing the Import/Export Tab” on page 71 to import or export members to your group.
- Click the Dynamic Group tab and continue to “Completing the Dynamic Group Tab” on page 72 to create extensions for your group.

Removing Members from the Group

To remove members from a Group:

1. Select the Static Group tab.
2. Click the Remove Members button.
3. When the Results table displays, enable the checkbox(es) to indicate which member(s) you want to remove from the group’s membership list, and then click the Remove Selected From Group button.

Figure 4-15 Removing A Member from the Group

Search For Members to Remove from Group

Basic Search **Advanced Search** Filtered Search

Enter the name of a User, Group, or Organizational Unit to search for:

Search type: User

Remove Selected From Group

Results Page: 1 of 1

	Name	First Name	Last Name
<input type="checkbox"/>	⊗ Bob Jones		
<input type="checkbox"/>	⊗ Googie Gocart		
<input checked="" type="checkbox"/>	Humphrey Gocart	Humphrey	Gocart
<input type="checkbox"/>	K. C. Jones	K. C.	Jones
<input type="checkbox"/>	Michael Miller	Michael	Miller

Page: 1 of 1

Directory Editor removes the selected member(s) from the group and returns you to the Static Group tab. Note that the Number of Members value has been updated to reflect the number of members currently in the group.

-
- NOTE**
- Directory Editor does not actually remove the members from the directory until you click the Save button on the Static Group page.
 - When you use the Remove Members feature to remove members in a group, it is possible to retrieve results for entries that do not really exist in the directory. This situation occurs if the `member` or `uniqueMember` attribute of `groupOfNames` or `groupOfUniqueNames` contains values that are not valid DNs.
 - A checkbox is displayed for invalid DNs. In addition, to denote a row with an invalid DN, Directory Editor displays the member name in red with an error icon.
-

4. You can proceed in several ways:
 - If you are finished removing members (and you have no other tasks to perform on this page), click Save to save your changes. Directory Editor returns you to the Browse page so you can verify your changes.
 - Click Add Members and return to “Adding Members to the Group” on page 67 to add members to the group.
 - Click Show Members and continue to “Showing Members in the Group” on page 71 to view the existing members of the group.
 - Click the Import/Export tab and continue to “Completing the Import/Export Tab” on page 71 to import or export members to your group.
 - Click the Dynamic Group tab and continue to “Completing the Dynamic Group Tab” on page 72 to create extensions for your group.

Showing Members in the Group

After creating and saving a new group, you can use the Show Members button to search for members within that group.

To view a list of current group members, use the following steps:

1. Click the Show Members button.

The Search For Members in Group page displays with the group members listed in a Results table located at the bottom of the page.

2. When you are finished, click OK to return to the Static Group page.

NOTE When you use the Show Members feature to search for group membership, it is possible to retrieve results for entries that do not really exist in the directory. This situation occurs if the member or uniqueMember attribute of groupOfNames or groupOfUniqueNames contains values that are not valid DNs.

To denote a row with an invalid DN, Directory Editor displays the member name in red with an error icon.

Completing the Import/Export Tab

When you are creating a new group, Directory Editor allows you to import group members using the Import/Export tab on the Group page.

Figure 4-16 Importing Members

The screenshot shows the 'Import/Export' tab selected in a window with other tabs: 'Quick Create', 'Static Group', 'Import/Export', and 'Dynamic Group'. The 'Import' section is active, displaying the text 'Import group members from a file.' and a note: 'The file should have the DN of each member to add to the group on it's own line.' Below this is a 'File:' label followed by a text input field and a 'Browse...' button. At the bottom of the section are two checkboxes: 'Import members not found in directory?' and 'Replace existing members?'. A dark 'Import Members' button is located at the bottom left of the form area.

When you import group members, Directory Editor expects a newline-delimited text file with a member DN on each line. (Directory Editor ignores empty lines and lines starting with the pound sign (#).)

To import members from a file:

1. Type the file name into the text box provided or click Browse to open a File Upload dialog box so you can locate and select the file name.
2. Enable one of the following checkboxes:
 - **To import members** (even if they are not found in the Directory Server), enable the Import members not found in directory? check box.
 - **To replace existing group members** with members in the file (instead of adding members in the file to the existing group members), enable the Replace existing members? check box. This feature is available for both group creates and edits. Directory Editor does not save import changes to the directory until you save the group.
3. When you are ready, click Import Members to import the file.
4. Click Save to save the new group members to your directory.

Completing the Dynamic Group Tab

You can use the Dynamic Group Tab to perform the following tasks:

- Create an optional Dynamic Group
- Add, edit, and remove LDAP search filters

Figure 4-17 Dynamic Group Tab

The screenshot shows the 'Dynamic Group' tab in a software interface. At the top, there are four tabs: 'Quick Create', 'Static Group', 'Import/Export', and 'Dynamic Group'. The 'Dynamic Group' tab is active. Below the tabs, there is a text input field with the label '* Name:' and the text 'Contractors' entered. To the right of this field is a legend: '* Indicates required field'. Below the name field is a section labeled 'Filters'. Inside the 'Filters' section, there is a button labeled 'Filter'. Below the 'Filters' section are three buttons: 'Add Filter', 'Edit Selected', and 'Remove Selected'.

Dynamic groups are often used to control access to applications and other data in the directory.

A dynamic group uses a set of LDAP search filters to determine membership in the group at any given moment. A dynamic group object consists of the `groupOfURLs` object class, a name (`cn`), and a multi-valued attribute (`memberURL`) containing all of the LDAP URLs that specify the search criteria.

These LDAP URLs contain information about which directory server to search, the scope of the search, the base context of the search, which attributes to return for the search, and the search filter.

For example, you might specify a dynamic group called *Contractors*, where the membership is determined by the `employeeType` equaling *Contractors*. For this example, the LDAP URL might be:

```
ldap://example.sun.com:389/ou=People,dc=com??sub?(employeeType=Contractors)
```

Instructions for adding, editing, and removing filters are provided in the following sections:

- “Specifying a Dynamic Group Name” on page 73
- “Adding Filters” on page 73
- “Editing Filters” on page 75
- “Removing Filters” on page 75

When you are finished with the Dynamic tab, you can

- Select another tab to provide additional information.
- Click Save to save the new information to the directory.
- Click Cancel to clear the text boxes.

Specifying a Dynamic Group Name

To create a dynamic group, select the Dynamic Group tab and enter a name for the new group in the Name text box.

Adding Filters

To add a filter for the dynamic group,

1. Select the Dynamic Group tab.
2. Click Add Filter and use the Advanced Search or Filtered Search tabs to configure a search filter.

NOTE Detailed instructions for using the Advanced Search and Filtered Search tabs are provided in Chapter 6, “Searching Directories.”

For example, to add a filter for the Contractors group, you might specify the following parameters on the Advanced Search tab:

- **Select Type:** User
 - **Base Context:** ou=People,dc=example,dc=com
 - **Search Conditions:**
 - **Attribute:** employeeType
 - **Condition:** contains
 - **Value:** contractor
3. When you are finished specifying the filter criteria, you can click the Test button to verify that your filter works as planned.

The results of your filter will display in a Results table. For example:

Figure 4-18 Example Results Table

The screenshot shows the 'Advanced Search' tab in a web interface. At the top, there are two tabs: 'Advanced Search' (selected) and 'Filtered Search'. Below the tabs, the 'Search type' is set to 'User'. The 'Base Context' is 'ou=People,dc=central,dc=sun,dc=' with a 'Browse' button next to it. Under 'Search Conditions', there is a table with one row: Attribute 'employeeType', Condition 'contains', and Value 'contractor'. Below this table are buttons for 'Remove Selected Attribute(s)', 'Add Attribute', 'Test', 'Save', and 'Cancel'. At the bottom of the search area are buttons for 'Create', 'Delete', 'Rename', 'Enable', and 'Disable'. Below these buttons is a header for the results: 'Results From Search of : ou=People,dc=central,dc=sun,dc=com'. To the right of the header is a pagination control: 'Page: 1 of 1 Go'. The results table has three columns: 'Name', 'First Name', and 'Last Name'. It contains two rows of data: 'Humphrey Gocart' (First Name: Humphrey, Last Name: Gocart) and 'K. C. Jones' (First Name: K. C., Last Name: Jones). At the bottom right of the results table is another pagination control: 'Page: 1 of 1 Go'.

4. If the test is satisfactory, click Save.

Editing Filters

To edit a filter, use the following steps:

1. On the Dynamic Group Form, enable the checkbox next to the filter you want to edit.
2. Click the Edit Selected button to edit the group filter.
3. When the Edit page is displayed, change the search parameters as necessary.
4. You can click Test to verify that your filter works as planned.
5. If you are satisfied with your changes, click Save to save the filter.

Removing Filters

To remove a filter, use the following steps:

1. On the Dynamic Group Form, enable the checkbox next to the filter(s) you want to remove.
2. Click the Remove Selected button to remove the filter.
3. Click Save.

Directory Editor will remove the filter and save the change to the directory.

Creating Organizational Units

Use the following steps to create a new organization:

1. From the Create tab, enable the Organizational Unit radio button.
2. Indicate where to add the new organization until by typing the parent's context information into the Parent Entry text field (or click Browse to locate the information). For example, you might enter:

```
ou=Groups,dc=example,dc=com
```

3. When you are finished, click Continue.

NOTE The Cancel button clears the Organization text box.

The Quick Create page is displayed (see Figure 4-19).

Figure 4-19 Quick Create Tab

The screenshot shows a window titled "Create" with "Save" and "Cancel" buttons in the top right. Below the title bar are four tabs: "Quick Create", "Common Attributes", "Locality", and "Other". The "Quick Create" tab is selected. Inside this tab, there is a sub-section titled "Common Attributes" with a legend: "* Indicates required field". Below this, there is a text input field labeled "* Organization Name:" containing the text "Engineering".

4. You can proceed in one of two ways:
 - The Organization Name is the only required attribute for this object type. Type a name in the text box provided, and then click Save.

Directory Editor saves the new organizational unit and the Browse page is displayed. The new name will be displayed in the tree.
 - After providing the required information, you can select one or more of the other tabs to provide additional information about the new object.

The following sections provide instructions for completing each tab on the Create page:

- “Completing the Common Attributes Page” on page 77
 - “Completing the Locality Page” on page 77
 - “Completing the Other Page” on page 79
5. When you are finished entering the new information on the Create page (being sure to provide all *required* information), click Save to add the new organizational unit to the directory.

Completing the Common Attributes Page

Use the Common Attributes page (Figure 4-20) to provide the following information about the new organizational unit:

- **Organization Name** (*required*): Enter a name for the new organization.
- **Description**: Enter a description of the organizational unit.
- **Business Category**: Enter a business category for the organization. For example, you might enter Product Development as an Engineering organization

Figure 4-20 Common Attributes Page

The screenshot shows a 'Create' dialog box with a title bar and 'Save' and 'Cancel' buttons. Below the title bar are four tabs: 'Quick Create', 'Common Attributes', 'Locality', and 'Other'. The 'Common Attributes' tab is selected. The main area contains three text input fields, each preceded by an asterisk and a label: '* Organization Name:' with the value 'Engineering', 'Description:' with the value 'Engineering staff', and 'Business Category:' with the value 'Product Development'. A legend in the top right corner states '* Indicates required field'.

Completing the Locality Page

Use the Locality page (Figure 4-21) to provide the following information about the new organizational unit:

- **Locality**: Enter the organization's location. For example, the city and state.
- **Post Office Box**: Enter the organization's post office box (if applicable).
- **Postal Address**: Enter the organization's street address.
- **Preferred Delivery Method**: Indicate how mail, packages, equipment, and so forth should be shipped to this organizational unit.
- **Office Name**: Enter the name of the facility where the organization is located.
- **Registered Address**: Enter the organization's registered address if the information is different from the postal address.

For example, if the new organization maintains offices in Austin, TX but is officially registered in Delaware, enter the Delaware address in this text box.

- **Street Address**: Enter the organization's street address.

- **State/Province:** Enter the state name or province name.
- **Search Guide:** Enter suggested search criteria if you are using the entry as the base object in the directory tree for a search operation.
- **See Also:** Enter another Directory Server entry that may contain information related to this entry.

Figure 4-21 Locality Page

The screenshot shows a 'Create' dialog box with a 'Locality' tab selected. The dialog has 'Save' and 'Cancel' buttons in the top right corner. Below the title bar are four tabs: 'Quick Create', 'Common Attributes', 'Locality', and 'Other'. The 'Locality' tab contains the following fields:

Locality:	Austin, TX
Post Office Box:	
Postal Address:	555 Desperado Lane
Postal Code:	55555-0000
Preferred Delivery Method:	UPS, Fed Ex
Office Name:	MyCompany, Inc
Registered Address:	
Street Address:	555 Desperado Lane, Building 3
State/Province:	Texas
Search Guide:	Austin
See Also:	Centex

Completing the Other Page

Use the Other Page to provide miscellaneous information about the new organizational unit, as follows:

- **Password:** Enter a password for the organization object.
- **Confirm Password:** Re-enter the password to confirm it.
- **Destination Indicator:** Enter the city and country associated with the entry needed to provide Public Telegram Service. (Generally used in conjunction with the `registeredAddress` attribute)
- **Facsimile Telephone Number:** Enter the organization's facsimile number.
- **International ISDN Number:** Enter the organization's ISDN (Integrated Services Digital Network) number.
- **Telephone Number:** Enter the organization's telephone number.
- **Teltext Terminal ID:** Enter the terminal identification number for a Teltext terminal (text telephone).
- **Telex Number:** Enter the organization's telex number.
- **x121 Address:** Enter the organization's Data Terminal Equipment (DTE) address to the company network.

Creating Domains

Use the following steps to create a new domain in the directory:

1. From the Create tab, enable the Domain radio button.
2. Indicate where to add the new domain by typing the parent's context information into the Parent Entry text field (or click Browse to locate the information). For example, you might enter:

```
dc=example,dc=com
```

3. When you are finished, click Continue (or press your Enter/Return key).

NOTE The Cancel button clears the Organization text box.

The Quick Create page is displayed. (see Figure 4-22).

Figure 4-22 Quick Create Page

4. You can proceed in one of two ways:
 - o The Domain Name is the only required attribute for this object type. Type a name in the text box provided (for example `testDC`), and then click Save.

Directory Editor saves the new domain name to the directory and the Browse page is displayed.
 - o After providing the required information, you can select one or more of the other tabs to provide additional information about the new domain.

The following sections provide instructions for completing each tab on the Create page:

- “Completing the Common Attributes Page” on page 81
 - “Completing the Locality Page” on page 81
 - “Completing the Other Page” on page 82
5. When you are finished entering the new information on the Create page (being sure to provide all *required* information), click Save to add the new domain to the directory.

Completing the Common Attributes Page

Use the Common Attributes page to provide the following information about your domain:

- **Domain Name** (*required*): Enter a name for the new domain.
For example, engineering.example.sun.com
- **Organization**: Enter a the name of the organizational unit to be associated with the new domain.
- **Description**: Enter a description of the new domain.

Completing the Locality Page

Use the Locality page to provide the following information about the new organizational unit:

- **Locality Name**: Enter the domain’s location.
- **Post Office Box**: Enter a post office box (if applicable).
- **Postal Address**: Enter a street address.
- **Postal Code**: Enter the zip or postal code of the domain.
- **Preferred Delivery Method**: Indicate how mail, packages, equipment, and so forth should be shipped to the organizational associated with this domain.
- **Office Name**: Enter the name of the facility where the domain is located.
- **Registered Address**: Enter the domain’s registered address if the information is different from the postal address.

For example, if the organization maintains offices in Austin, TX but is officially registered in Delaware, enter the Delaware address in this text box.

- **Street Address**: Enter the domain’s street address.

- **State/Province:** Enter the state name or province name.
- **Search Guide:** Enter suggested search criteria if you are using the entry as the base object in the directory tree for a search operation.
- **See Also:** Enter another Directory Server entry that may contain information related to this entry.

Completing the Other Page

Use the Other Page (Figure 4-23) to provide miscellaneous information about the new domain.

Figure 4-23 Other Page

The screenshot shows a 'Create' dialog box with four tabs: 'Quick Create', 'Common Attributes', 'Locality', and 'Other'. The 'Other' tab is selected and contains the following fields:

- Associated Name:
- Password:
- Confirm Password:
- Business Category:
- Destination Indicator:
- Facsimile Telephone Number:
- International ISDN Number:
- Telephone Number:
- Teletex Terminal Id:
- Telex Number:
- X121 Address:

'Save' and 'Cancel' buttons are located in the top right corner of the dialog.

Complete this page as follows:

- **Associated Name:** Enter a name in the organizational directory tree that is associated with a DNS domain.
- **Password:** Enter a password for the domain object.
- **Confirm Password:** Re-enter the password to confirm it.
- **Business Category:** Enter a business category for the domain. For example, you might enter Product Development for an Engineering-related organization.

- **Destination Indicator:** Enter the country and city associated with the entry needed to provide Public Telegram Service. (Generally used in conjunction with the `registeredAddress` attribute)
- **Facsimile Telephone Number:** Enter the domain's facsimile number.
- **International ISDN Number:** Enter the domain's ISDN (Integrated Services Digital Network) number.
- **Telephone Number:** Enter the domain's telephone number.
- **Teltext Terminal ID:** Enter the terminal identification number for a Teltex terminal (text telephone).
- **Telex Number:** Enter the domain's telex number.
- **x121 Address:** Enter the domain's Data Terminal Equipment (DTE) address to the company network.

Creating Localities

Use the following steps to create a new locality:

1. From the Create tab, enable the Locality radio button.
2. Indicate where to add the new locality by typing the parent's context information into the Parent Entry text field (or click Browse to locate the information). For example, you might enter:
`dc=example,dc=com`
3. When you are finished, click Continue (or press your Enter/Return key).

NOTE The Cancel button clears the Organization text box.

The Quick Create page is displayed (see Figure 4-24).

Figure 4-24 Quick Create Tab

The screenshot shows a 'Create' dialog box with two tabs: 'Quick Create' and 'Common Attributes'. The 'Common Attributes' tab is active. A legend in the top right corner states '* Indicates required field'. Below the legend, the text 'Common Attributes' is displayed. A single text input field is labeled '* Locality Name:' and contains the text 'North America'.

4. You can proceed in one of two ways:
 - The Locality Name is the only required attribute for this object type. Type a name in the text box provided, and then click Save.
Directory Editor saves the new locality name and the Browse page is displayed. The new Locality name will be displayed in the tree.
 - After providing the required locality name, select the Common Attributes tab to specify additional information about the locality. (Go to Step 1.)
5. Select the Common Attributes tab.
6. Type the following information in the text boxes provided:
 - **Locality Name** (*required*): Enter a name for the new locality.

NOTE Directory Editor automatically completes this field if you provided a Locality Name on the Quick Create tab.

- **Street Address:** Enter the street and city address for this locality.
- **State/Province:** Enter the state or province for this locality.
- **Description:** Enter a description of the locality.
- **Search Guide:** Enter suggested search criteria if you are using the entry as the base object in the directory tree for a search operation.
- **See Also:** Enter another Directory Server entry that may contain information related to this entry.

For example, you might enter the following information for a North American sales office:

Figure 4-25 Locality Common Attributes Page

The screenshot shows a 'Create' dialog box with two tabs: 'Quick Create' and 'Common Attributes'. The 'Common Attributes' tab is active. The form contains the following fields:

- * Locality Name: North America
- Street Address: 333 Casa Blanca Cove, Austin
- State/Province: TX
- Description: North American Sales Headquarters
- Search Guide: sales headquarters
- See Also: sales

* Indicates required field

7. When you are finished with this tab, you can
 - o Click Save to save the new information to the directory and return to the initial Create page.
 - o Click Cancel to clear the text boxes and return to the initial Create page.

Editing Objects

-
- NOTE**
- Most of the procedures for editing an object's attributes are the same as those you used to create the object. Review the instructions provided in "Creating Objects" on page 49 if necessary.
 - If you are editing an object with attributes that are not in the schema for the object's object classes, Directory Editor will display those attributes on the *Other* tab.

To move these attributes to a different tab, you must edit the form for that object class.
 - After a user logs in, Directory Editor will not allow them to delete, disable, or rename the object (dn) they used to login. If they try to change the object an error message will result.
-

This section explains how to access the Edit pages and provides instructions for enabling/disabling, renaming, and deleting objects using Directory Editor. The information is organized as follows:

- "Accessing an Edit Page" on page 86
- "Enabling or Disabling Objects" on page 91
- "Renaming Objects" on page 93
- "Deleting Objects" on page 95

Accessing an Edit Page

You can access the Edit pages, using one of the following methods:

- From the Home page, click the Edit My Directory Information link to edit your information in the directory.

NOTE The Edit My Directory Information link does not display unless the user who is logged in has write access to their entry.

An Edit page is displayed so you can edit the following information:

- Common Attributes (review page 52)
- Group Membership (review page 54)
- Location Information (review page 55)
- Misc Attributes (review page 56)
- Extensions (review page 57)
- Select the Browse tab, enable a checkbox to the left of a node in the Browse tree.

Figure 4-26 Enable the Checkbox in the Browse Tree



The buttons located above the Browse tree become active. Use these buttons to perform any of the following tasks:

- Edit the object's attributes
- Enable or Disable the object (see page 91 for instructions)
- Rename the object (see page 93 for instructions)
- Delete the object (see page 95 for instructions)
- Select the Browse tab and click the linked name of the node you want to edit.

Figure 4-27 Click the Linked Name



- Select the Search tab, search for the object you want to edit, and click the linked name of the object you want to edit in the Results table (see Figure 4-28).

Figure 4-28 Enable the Object in the Results Table

Results From Search of : dc=central,dc=sun,dc=com			
Page: 1 of 2 Go			
<input type="checkbox"/>	Name	First Name	Last Name
<input type="checkbox"/>	K. C. Jones	K. C.	Jones
<input type="checkbox"/>	Linus Larrabee	Kenneth	Suter
<input type="checkbox"/>	Nick	Nicholas	Charles
<input checked="" type="checkbox"/>	Nora	Nora	Charles
<input type="checkbox"/>	Rosie Sayer	Rose	Sayer

Page: 1 of 2 Go

Either method opens the Edit an Object page (shown in Figure 4-29):

Figure 4-29 Example Edit an Object Page

Edit an Object
Save Cancel

Common Attributes
Group Membership
Location Info
Misc Attributes
Extensions

* Indicates required field

Object: cn=Michael Miller,ou=People,dc=central,dc=sun,dc=com

User Name:

* First Name:

* Last Name:

Common Name/Nickname: ⊕ Add

Password:

Confirm Password:

Email: ⊕ Add

Telephone: ⊕ Add

Department #:

Photo URL:

Employee #:

NOTE If you are editing an object and remove an object class from the Extensions tab, Directory Editor will remove all attributes belonging to the removed object class when you save the object.

Importing and Exporting Group Members

When you edit a group, Directory Editor allows you to import and export group members using the Import/Export tab on the Group form.

Figure 4-30 Import/Export Tab

The screenshot shows a window with four tabs: "Default Dynamic Group Form", "Static Group", "Import/Export", and "Extensions". The "Import/Export" tab is active. It contains two main sections: "Export" and "Import".

Export Section:

- Text: "Export the group members to a file."
- Button: "Download Members"

Import Section:

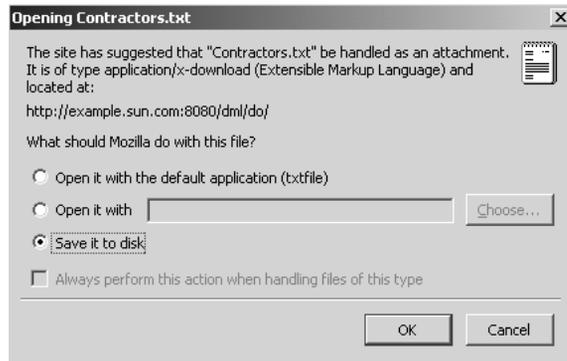
- Text: "Import group members from a file."
- Text: "The file should have the DN of each member to add to the group on it's own line."
- Form: "File:" followed by a text input field and a "Browse..." button.
- Checkboxes:
 - Import members not found in directory?
 - Replace existing members?
- Button: "Import Members"

Exporting Group Members

When you export group members, you download all of the member DNs to a newline-delimited text file.

To export group members:

1. Select the Import/Export tab and then click the Download Members button.
2. When the Opening <group_name>.txt dialog box is displayed, (Figure 4-31), enable one of the radio buttons to indicate how your browser should handle the file.

Figure 4-31 Specifying How Your Browser Should Open the File

- Open it with the default application (for example, `txtfile`)
 - Open it with (You must enter an application name in the text field provided.)
 - Save it to disk (You must specify a file name and specify a folder in which to save the file.)
3. After making your selection, click OK to close the dialog box.
 4. Click Save to save the exported group file to your directory.

You should now be able to open the `.txt` file in any standard text editor (such as Microsoft[®] Notepad). At this point, you can edit the file, and then import it using the instructions provided in the next section, or store the file for backup purposes.

Importing Group Members

When you import group members, Directory Editor expects a newline-delimited text file with a member DN on each line. (Directory Editor ignores empty lines and lines starting with the pound sign (#).)

To import members from a file:

1. Type the file name into the text box provided or click **Browse** to open a File Upload dialog box so you can locate and select the file name.
2. Enable one of the following checkboxes:
 - **To import members** (even if they are not found in the Directory Server), enable the **Import members not found in directory?** check box.
 - **To replace existing group members** with members in the file (instead of adding members in the file to the existing group members), enable the **Replace existing members?** check box. This feature is available for both group creates and edits. Directory Editor does not save import changes to the directory until you save the group.
3. When you are ready, click **Import Members** to import the file.
4. Click **Save** to save the new group members to your directory.

Enabling or Disabling Objects

You use the **Enable** and **Disable** buttons to activate or inactivate a user account for the Sun Directory Server. Once inactivated, a user cannot bind to the directory and the authentication operation will fail.

NOTE When you create an object, it is enabled in the directory by default. However, if you disabled the object at some point, you may need to re-enable it again.

2. You are prompted to confirm the action.
 - Click Enable or Disable (whichever is appropriate) to continue.
 - Click Cancel to return to the Search page without making any changes to the object.

Renaming Objects

To rename an object, use the following steps:

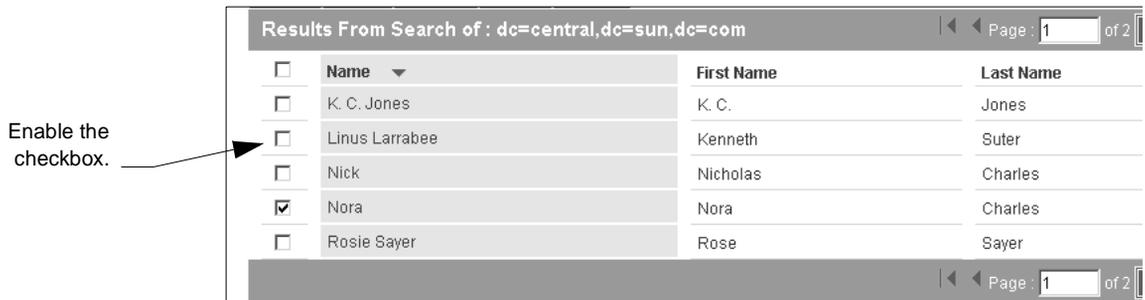
1. Open the Rename page using one of the following methods:
 - Select the Browse tab, enable a checkbox to the left of the object in the Browse tree, and then click the Rename button.

Figure 4-35 Enable the Checkbox in the Browse Tree



- Select the Search tab, search for the object to edit, enable the checkbox to the left of the object name in the Results table, and then click the Rename button.

Figure 4-36 Enable the Object in the Results Table



The Rename page is displayed as shown in the following figure:

Figure 4-37 Rename Page

? Rename

Existing DN: cn=Sam,ou=People,dc=central,dc=sun,dc=com

New DN:

Rename Cancel

2. Enter a new DN name in the text box provided.
3. Click the Rename button.

Figure 4-38 Example Rename Page

? Rename

Existing DN: cn=Sam,ou=People,dc=central,dc=sun,dc=com

New DN: cn=Sam Spade,ou=People,dc=example,dc=sun

Rename the children of this object?

4. If the object has children, Directory Editor provides a checkbox and asks if you want to rename the children. Enable this checkbox to rename the children.

Directory Editor renames the object (and children) and returns you to the Browse page so you can verify the change.

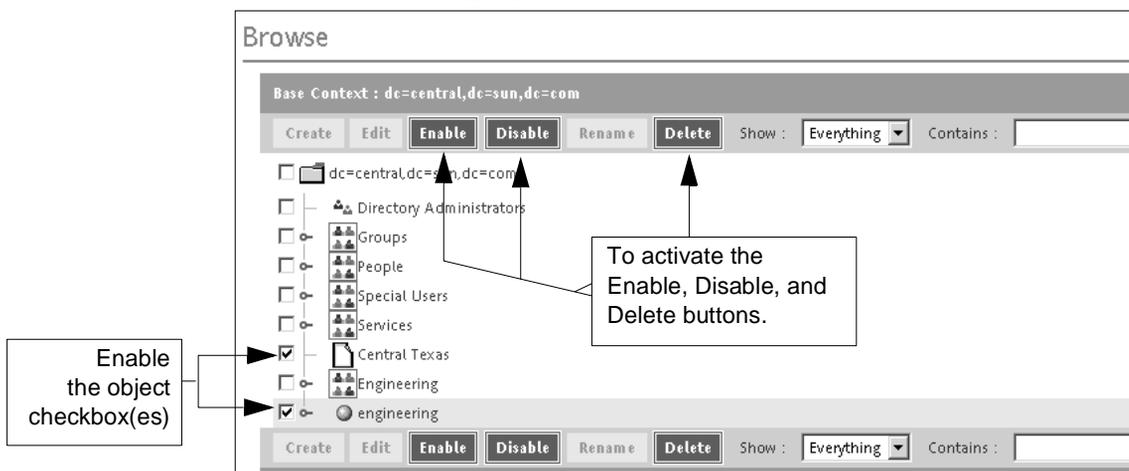
Deleting Objects

If you want to delete an object you have created, use the following procedure:

1. Select the Browse tab.
2. Enable the checkboxes located to the left of the object(s) you want to delete.

The Enable, Disable, and Delete buttons located above the Browse tree will become active.

Figure 4-39 Deleting Objects



3. Click the Delete button.

The Delete page is displayed. You are asked to confirm whether to delete the selected object(s) and the children of the selected object.

4. When you are ready, click the Delete button to delete the selected object(s).

Directory Editor immediately removes the object (and children if specified) and then returns you to the Browse page.

Working With Forms

This chapter explains how to customize the default Directory Editor forms and how to create new forms for your enterprise. The information is organized as follows:

- “Overview” on page 97
- “Accessing the Forms Page” on page 101
- “Customizing Forms” on page 103
- “Creating New Forms” on page 115
- “Deleting Customized Forms” on page 118
- “Testing Your Form” on page 118

Overview

Directory Editor provides the following forms, which you can customize by incorporating business logic and manipulating data before presenting them to your end users:

Table 5-1 Directory Editor Forms

Form Name:	Use this form to:
applicationProcess	Edit Directory Editor configuration objects.
Default Browse Form	List the managed directory's contents
Default Create Form	Create users, groups, or organizations
Default Delete Form	Delete users, groups, or organizations
Default Disable Form	Disable specific objects

Table 5-1 Directory Editor Forms (*Continued*)

Default Enable Form	Enable specific objects
Default Home Form	Configure a Home page
Default Organization Picker Form	Select organizations
Default Rename Form	Rename objects
domain	Define domains for the directory
extensibleobject	Define attribute values for a class of service (COS) template.
groupofnames	Create a list of email distribution aliases.
groupofuniquenames	Add members to or remove members from a group, rename groups, search groups, export DNs to a text file, or display group memberships.
groupOfURLs	Create and edit dynamic groups.
inetorgperson	Control user attributes and to which groups an end-user will belong.
locality	Define localities for the directory
organizationalperson	Create and define user attributes for organizational members
organizationalunit	Create and define attributes for organizational units

-
- NOTE**
- You cannot edit the *standard* forms (those forms beginning with “Default”), but you can create a customized version of these forms by clicking the Customize button to open the form editor.
 - For information about the specific HTML components that define the default Directory Editor forms, see “Appendix A, “HTML Components Used to Define Directory Editor Forms.”
 - If Directory Server does not have a schema entry for an object, and you want to create a form for that object, you must add the required object classes to the schema. Directory Editor reads the schema entries to build the form fields.
-

How Do Forms Work?

Directory Editor forms contain rules that govern how the browser displays user view attributes on a application's web page. These forms control:

- **Page layout and display characteristics.** Controls whether fields are visible, read-only, or hidden from view.

Visible field types include simple text boxes, radio buttons, and selection boxes with multiple values. Fields values also can be based on other fields.
- **How data is used on a page.** Captures data dynamically from a resource (such as a Directory Server) or calculates the data from other fields.
- **How data comes into the system.** Interfaces with web pages or non-interactive systems. In this role, the form has no visual fields, but still provides rules to set default values and other field values.

For example, the Full Name field may not be visible to an administrator using the page, but the field value can be set based on values the end-user enters into the First Name, Middle Name, and Last Name fields.

Populating fields from other fields reduces the amount of data entry that users and administrators must perform, consequently reducing potential data entry errors. Similarly, by providing option menus instead of text input fields, an administrator can select a department from a list instead of having to type the department name.

Various factors affect how the browser displays a form. However, form behavior within a browser is primarily determined by how display components are used in the form. You can associate form fields with a display component that determines how the field is displayed in the browser.

Why Edit the Forms?

Why customize the default Directory Editor forms, which already provide all the fields that you need to perform actions within the product? Customizing the default forms allow you to better enforce your company's policies and processes:

- Preserve privacy by limiting the amount of user account information displayed on the screen.
- Omit nonessential information to reduce distractions for users performing a specific task. Display only the fields needed to accomplish the current task.

Customizing the default fields in forms allows you to extend and customize the application for your environment. Specifically, you can customize the default forms to:

- Address the specific needs of the users in your organization.

This type of customization is particularly important when different types of administrators must access user data, but it is not appropriate for all of the administrators to have access to all of the user account attributes. For example, a human resources administrator will need to view different subset of user account attributes than a help desk administrator.

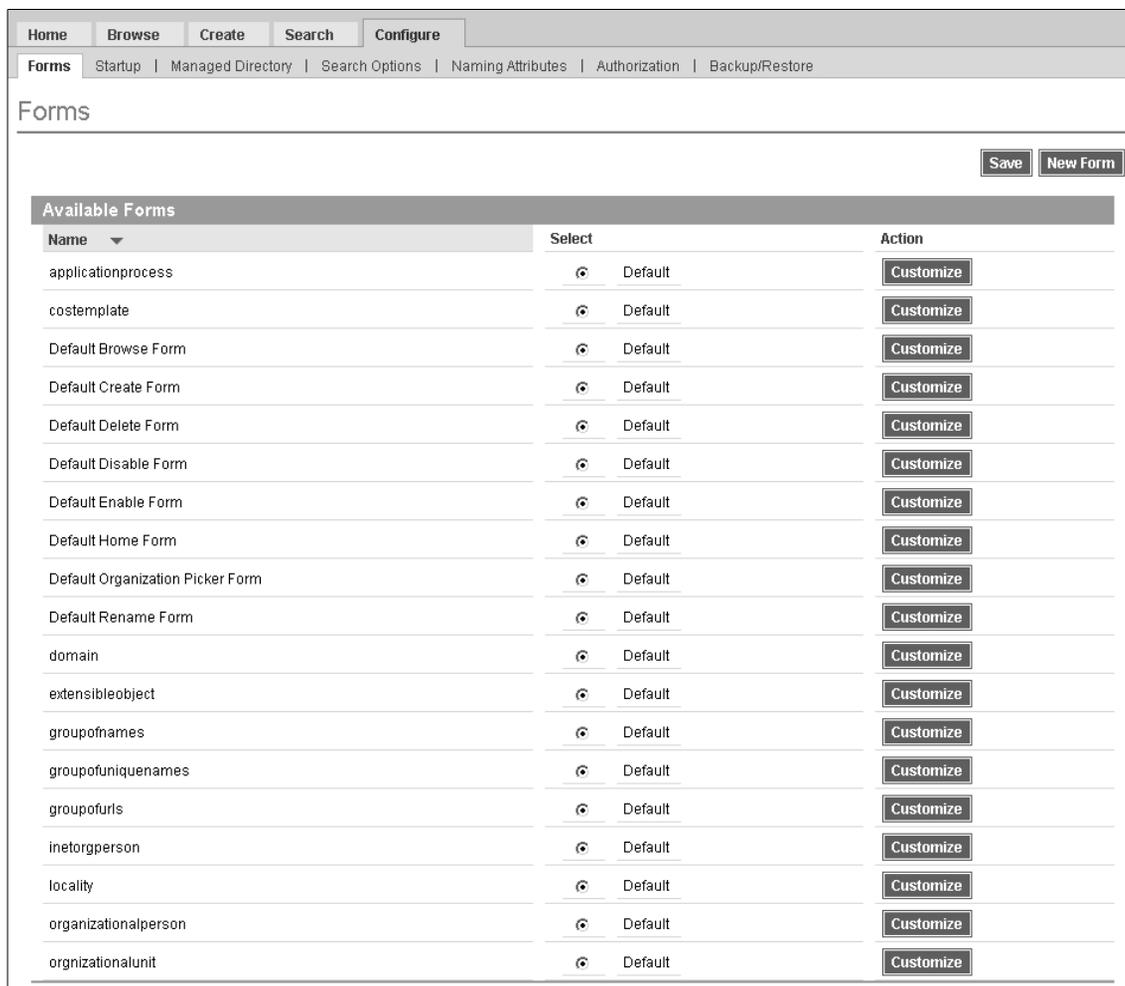
NOTE Directory Editor does not display attributes to users who do not have read access for those attributes. In addition, Directory Editor will display read-only attributes as read-only fields.

- Control the display and content of the user account attributes, particularly attributes displayed on the Create and Edit pages. These pages contain most of the attributes that need to be controlled.
- Define default values for user view attributes and their associated attributes.
For example, you could define a default account name for a user based on a combination of first and last initials and their employee ID number instead of the administrator having to key in the information.
- Post-process user view attributes data entry.
For example, you can automatically create a mail account based on the value of a location field.
- Control screen real estate by positioning multiple fields on one line.
By rearranging fields, you can make the form more closely resemble a printed form or pre-existing web form.
- Define rules for the way hidden attributes are calculated.
For example, a user's email address can be calculated to be the user's first name, a period, their last name, then the mail domain: *joe.user@example.com*.
- Correspond to the Directory Server environment.
You can create new forms to allow creating and editing objects that are unique to the Directory Server's schema.

Accessing the Forms Page

If you want to customize or design a form, select the Configure tab, and then click the Forms tab. (See the following figure.)

Figure 5-1 Forms Tab



This page contains:

- **Name column:** Lists all of the default forms provided with Directory Editor. (Use the scroll bar to view all of the forms in the Name column.)

- **Select column:** Contains radio buttons that are all labeled *Default*.

After you customize any of the default forms, Directory Editor adds a *Custom* radio button to this column. Changing this radio button controls whether Directory Editor uses the Default or Custom form.

- **Action column:** Contains Customize, Edit, and/or Delete Customized buttons. Click these buttons to customize the selected form.

After you customize any of the default forms, Directory Editor changes the Customize button to an *Edit* button and adds a *Delete Customized* button to this column. You can click the Delete Customized button to remove a customized form. (You cannot delete a default Directory Editor form.)

Figure 5-2 Action Column

Select		Action		
<input checked="" type="radio"/>	Default	Customize		
<input checked="" type="radio"/>	Default	Customize		
<input checked="" type="radio"/>	Default	Customize		
<input type="radio"/>	Default	<input checked="" type="radio"/> Custom	Edit	Delete Customized
<input checked="" type="radio"/>	Default	Customize		

- **Save button:** Click to preserve changes to the Default or Custom radio buttons.
- **New Form button:** Click to create a new form.

Instructions for using this page to customize the default Directory Editor forms or to create new forms for your site are provided in the following sections:

- “Customizing Forms” on page 103
- “Creating New Forms” on page 115

Customizing Forms

This section provides instructions and examples for customizing forms, and is organized as follows:

- “Opening a Form” on page 103
- “Working with the Forms Page” on page 104
- “Performing Common Tasks” on page 107

Opening a Form

To open a form for customizing, click the Customize or Edit button in the Action column on the Available Forms page.

A Forms page is displayed, which consists of two views:

- A *tree-view* list of all elements associated with that form on the left
- A *property view* with a graphical representation of the form on the right.

For example, if you decided to customize the Default Create Form, the following page would display:

Figure 5-3 Default Create Form

The screenshot shows a window titled "Forms" with a tree view on the left and a property view on the right. The tree view shows a folder "Form" containing "New Object Type:", "Parent Entry:", and "buttons". The property view shows "New Object Type:" with radio buttons for "User", "Group", "Organizational Unit", "Domain", and "Locality". Below it is a "Parent Entry:" text field with a "Browse" button. At the bottom left are "Save", "New Field", and "Cancel" buttons. At the bottom right are "Continue" and "Cancel" buttons. A note "* Indicates required field" is in the top right of the main area.

When you select elements in the Forms tree to edit, the property view interface changes so you can edit the properties associated with that element. For more detailed information, continue to the next section.

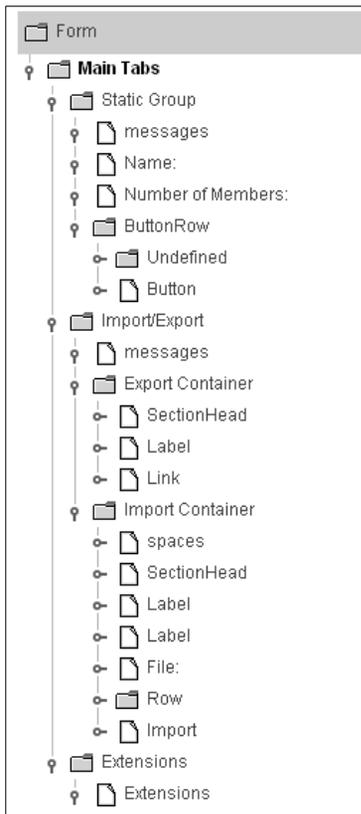
Working with the Forms Page

This section provides instructions for working with the Forms page.

Working in Tree View

The tree view (located on the left side of the Forms page) shows a hierarchical view of the form elements. The tree lists each element in order, nesting children under their parent.

Figure 5-4 Tree View Display (Expanded)



The node icons used in the tree are based on whether the form has children, and include the following:

-  The folder icon displays if the form item has children.
-  The page icon displays if the form item does not have children.

The tree also contains the following icons:

- Click the key icon  to view any nested children in that form item node.
- Click this key icon  to collapse the form item.

To customize the properties associated with each form or field element, click the node label, and the right side of the Forms page changes as follows:

- Click Form  Form to specify a name, description, title, and/or a subtitle for the form (see “Identifying the Form” on page 107).
- Click any other named element  Select type to specify the field properties and location (see “Customizing Field Properties” on page 109 and “Changing Field Locations” on page 112).
- Click the New Field button to create a new field for the form (see “Creating New Fields” on page 114).

Working in Property View

The property view (located on the right side of the Forms page) provides a graphical representation of the form you are customizing.

When you initially open a form for editing, the property view displays the form as it is currently configured. However, when you select an element in tree view, the property view changes so you can edit properties associated with that element.

For example, if you select the Default Create Form, the initial property view is shown in Figure 5-5.

Figure 5-5 Initial Property View: Create Form

* Indicates required field

* New Object Type: User Group Organizational Unit Domain Locality

* Parent Entry:

Then, if you select the New Object Type node in tree view, the property view changes as shown in Figure 5-6.

Figure 5-6 Property View: Editing New Object Type Element

Field DML_prop_label_type

Field Properties | Move Field

Name:

Title: Options: Required

Display Properties

Type:

Allowed Values:

Allowed Values Expression:

Labels:

Options: No New Row

When you are editing elements, the property view contains two tabs:

- **Field Properties tab:** Provides a schematic representation of the form and enables you to customize properties associated with that element.
- **Move Field tab:** Enables you to control the location of different form elements.

After changing form elements in property view, if you select the Form node in tree view, the property view changes to reflect your changes (if your changes affected the visible features of the form).

NOTE Remember, this view is only a preview of the changes made to your form — not a working form.

Performing Common Tasks

This section describes how to perform the common tasks associated with customizing Directory Editor forms. The information is organized as follows:

- “Identifying the Form” on page 107
- “Customizing Field Properties” on page 109
- “Changing Field Locations” on page 112
- “Creating New Fields” on page 114
- “Deleting Fields” on page 114
- “Saving Changes” on page 115

Identifying the Form

To identify your customized form, you can specify a name, description, title, and/or subtitle as follows:

1. If the Form node is not already selected, click the Form folder icon (top node in the Forms tree) and the page changes as follows:

Figure 5-7 Identifying a Default Create Forms

The screenshot shows a dialog box titled "Forms". On the left, there is a tree view with a "Form" folder icon selected. Below it are three sub-items: "Select type:", "Organization", and "buttons". At the bottom of the tree view are three buttons: "Save", "New Field", and "Cancel". On the right side of the dialog, there are four text input fields labeled "Name", "Description", "Title", and "Sub-title". At the bottom right of the dialog is an "OK" button.

2. Complete one or more of the following fields (all optional):
 - **Name:** Enter a name for this form.
 - **Description:** Enter a meaningful description of the form.
 - **Title:** Enter the text you want to display at the top of the form. For example, *MyCompany Create Form*.
 - **Subtitle:** Enter the text you want to display beneath the form's title. For example, you might indicate the form's purpose: *Use this form to create new users, groups, and organizational units.*

3. When you are finished, click OK.

The Form node in tree view changes to use the Title you specified, and the new form displays on the right side of the page. For example:

Figure 5-8 New Form Page

The screenshot shows a window titled "Forms" with a tree view on the left and a form configuration area on the right. The tree view shows a folder "My Company Create Form" which is selected, containing three items: "New Object Type:", "Parent Entry:", and "buttons". Below the tree view are buttons for "Save", "New Field", and "Cancel". The main area is titled "My Company Create Form" and contains the text "Use this form to create new users, groups, and organizational units." Below this is a legend: "* Indicates required field". The form configuration area includes:

- * New Object Type: User Group Organizational Unit Domain Locality
- * Parent Entry:

 At the bottom right of the main area are buttons for "Continue" and "Cancel".

4. Continue as follows:
 - o If you have other changes to make to the current form, select another node in the tree.
 - o If you want to add one or more new fields, click the New Field button and proceed to "Creating New Fields" on page 114.
 - o If you have no other changes to make, click Save.

Customizing Field Properties

To customize properties for a particular field on your form, use the following steps:

1. In the Form tree, click the field element you want to customize.
For example, click Select type and the following page is displayed:

Figure 5-9 Customizing Field Properties

The screenshot shows a software interface for customizing field properties. On the left, a tree view under 'Forms' shows a folder 'MyCompany Create Form' containing 'New Object Type:', 'Text', and 'buttons'. Below the tree are 'Save', 'New Field', and 'Cancel' buttons. The main area is titled 'Field New Object Type:' and has two tabs: 'Field Properties' (selected) and 'Move Field'. The 'Field Properties' tab contains the following elements:

- Name:** primaryObjectClass
- Title:** New Object Type: **Required**
- Display Properties:**
 - Type:** Select (dropdown menu)
 - Null Value:** (text input)
 - Size:** (text input)
 - Options:** Sorted Multiple No New Row
- Allowed Values:** A list box containing: inetOrgPerson, organizationalPerson, pers, groupOfUniqueNames, top, organizationalUnit, top, domain, top, locality, top.
- Allowed Values Expression:** (text input)

At the bottom right of the dialog are 'OK' and 'Delete' buttons.

The content of the Field Properties tab will change, depending on which field element you select. The following table lists all of the different Field Properties parameters you might encounter as you work.

Table 5-2 Field Properties

Property	Description
Name	Type a name for this field. <ul style="list-style-type: none"> You must enter a name that specifies a view path for all fields that display as editing components (such as text boxes, checkboxes, and selects). You do not have to name fields that do not display as editing components (such as <code>SectionHead</code> or <code>Javascript</code>); however, you can name non-editing fields if they are referenced by another form through a Field reference. You must use the following format to name fields that correspond to LDAP attributes: <code>attributes[<attribute_name>].value</code> For example, use the following format to name a cn attribute: <code>attributes[cn].value</code>
Title	Specify text to display adjacent to the field.
Options:	<p>Required checkbox: Enable this checkbox if the field is required to process the form. If you enable this option:</p> <ul style="list-style-type: none"> The user must enter a non-null value in this field before submitting the form. A red asterisk is displayed to the left of the field and a message displays at the top of the form to explain that a red asterisk denotes required fields.
Display Properties:	
Type	Use the drop-down menu to select which type of display element you want to use for the new field. For example, do you want to create a text box or a button?
Available Title	Enter a title for the available values of a multi-select object.
Selected Title	Enter a title for the selected values of a multi-select object.
Allowed Values	Use this text box to enter allowed values for the component. The end-user can select only those values provided in this list. (Some values may already be displayed by default.)
Allowed Values Expression	Use an expression to create a list of allowed values for the component. For convenience when setting properties in XML forms, you can specify the allowed values as a comma-delimited list.
Size	Specify the character width of the control (for text boxes).
Maximum Length	Specify the maximum number of characters that can be entered in a text box.
Display Name	Specify text to display next to a check box.
Rows/Columns	Specify the number of rows and columns to display in a text area.
Null Label	Specify a label to display in select fields to indicate that nothing is selected.
Labels	Specify radio button labels.

Table 5-2 Field Properties (*Continued*)

Show Child Count	Enable this checkbox to see how many children are under each node in the Forms tree. The number will be displayed in brackets after the node name. For example, People [21].
Show Only Branches	Enable this checkbox to display just the branches in the Forms tree instead of the branches and leaves. For example, if you open a branch that only has leaf nodes in the Browse dialog — you will not see anything; however, on the Browse tab you will see all the leaves.
Allow Multiple Selections	Enable this checkbox to allow the user to select multiple elements in the Forms tree.
Yoke to Active Mode	Enable this checkbox to allow the window to automatically scroll to the last node with which you interacted. Normally, if you are working with a large Forms tree, and you expand a branch node that you had to scroll down to see, when you are done with the expansion you would have to reset the tree to the top. However, if you enable this property, the tree will automatically scroll the window to the expanded node.
Display Page Controls On Branch Node	Enable this checkbox to allow the page navigation tool to display when browser entries exceed the maximum number of lines allowed for that page. (See page 43 for a description of the page navigation tool.)
Use Paged Results	Enable this checkbox to allow Directory Editor to add pages when browse or search results exceed the display limit allowed for one page.
Paged Results Size	Enter a value to specify a maximum number of entries that can be displayed on a single page. Default is five entries.
Number of pages before search link is added	Enter a value to specify how many pages can exist before Directory Editor provides a search link on the page.
Maximum number of unpagged results	Enter a value to specify how many unpagged results can exist before Directory Editor provides a warning message.
Options:	Enable one or more of the following checkboxes — as necessary: <ul style="list-style-type: none"> • Left Label: Forces the label to display to the left of a field. • No New Row: Forces the field to appear to the right of the proceeding field. For example, Name fields are examples where this is useful, where it is desirable to allow the user to enter the last name, first name, and middle initial from right to left, rather than down the page. • Multiple: Controls whether multiple selections are available for a select field. • Multi-Valued: Allows multiple values for a text field. • Ordered: Controls whether multi-select values can be moved up and down in the selected list. • Read-Only: Creates a non-editable field. • Secret: Causes typed text to display as asterisks (*). (typically used for encrypted data, such as passwords) • Sorted: Controls whether select and multi-select values are sorted alphabetically.

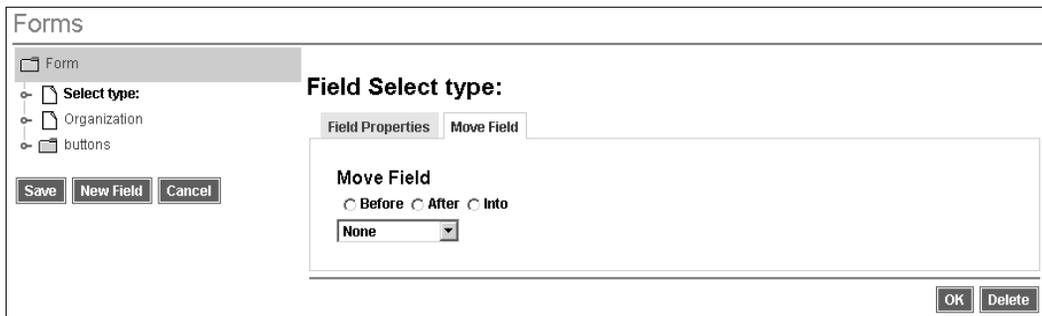
2. On the Field Properties tab, you can
 - Customize one or more of the property parameters described in the preceding table, and then click OK.
 - Click the Delete button (located at the bottom of the tab) to remove a field from this form. Repeat to delete additional fields.
3. When you are finished customizing the properties for this field, you can:
 - Select the Move Field tab to change the field's location on this form. (Proceed to "Changing Field Locations" on page 112.)
 - Add or remove a field (Proceed to "Creating New Fields" on page 114 or "Deleting Fields" on page 114 respectively.)
 - Select a different field to customize that field's properties.
 - Save or Cancel your changes and return to the Available Forms page. (Proceed to "Saving Changes" on page 115.)

Changing Field Locations

Use the following steps to change the location of a particular field on your form:

1. In the Form tree, click the field element you want to move.
2. When the Properties tabs are displayed, select the Move Field tab.

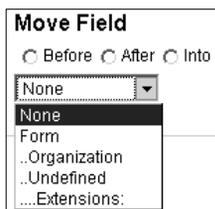
Figure 5-10 Move Field Tab



The Move Field tab contains three radio buttons and a drop-down menu. Elements displayed in the drop-down menu are indented to indicate whether they are tabs, fields, or nested children.

For example, in the following figure, *Form* is the form, *Organization* and *Undefined* are field elements, and *Extensions:* is a child of *Undefined*.

Figure 5-11 Move Field Menu



NOTE You can only move field elements and children *into* the Form.

3. Decide where to move the selected field, enable one of the following buttons, and then make a selection from the drop-down menu:
 - **Before button:** Moves the selected field element above a field you select from the drop-down menu.
 - **After button:** Moves the selected field element below a field you select from the drop-down menu.
 - **Into button:** Moves the selected field element into a tab you select from the drop-down menu.
4. When you are finished with the Move Field tab, you can:
 - Select a different field to customize.
 - Add or remove a field (Proceed to “Creating New Fields” on page 114 or “Deleting Fields” on page 114 respectively.)
 - Save or Cancel your changes and return to the Available Forms page. (Proceed to “Saving Changes” on page 115.)

NOTE You must click the Save button located under the Form tree to save your change(s) to the directory.

Creating New Fields

To Create a new field for your form:

1. Click the New Field button, located under the Form tree.

Directory Editor adds an undefined node to the Form tree, and the property view changes as follows:

Figure 5-12 Creating a New Field

The screenshot shows the Directory Editor interface. On the left, a tree view under 'Forms' shows a 'Form' folder containing 'Select type:', 'Organization', and two 'Undefined' nodes. Below the tree are 'Save', 'New Field', and 'Cancel' buttons. The main area is titled 'Field Undefined' and has two tabs: 'Field Properties' (selected) and 'Move Field'. The 'Field Properties' tab contains:

- 'Name' and 'Title' text input fields.
- 'Display Properties' section with a 'Type' dropdown menu set to 'None'.
- 'Options' section with a 'Required' checkbox.

 At the bottom of the dialog are 'OK' and 'Delete' buttons.

2. Complete the Field Properties tab as described in “Customizing Field Properties” on page 109.
3. Use the Move Field tab to position the new field on your form as described in “Changing Field Locations” on page 112.
4. When you are finished, click Save save your change(s) to the directory.

Deleting Fields

To Delete a field element from your form, click the Delete button that is located at the bottom of the Field Properties tab. Directory Editor will remove that field element from the form (and tree view).

NOTE You must click the Save button located under the Form tree to save your change(s) to the directory.

Canceling Changes

If you make a mistake while customizing a form or decide you do not want to use the form you customized, click the Cancel button located at the bottom of the Form tree.

Directory Editor will cancel any unsaved changes made to that form and immediately return you to the Available Forms page.

Saving Changes

You can save your customization changes at any time by clicking the Save button located at the bottom of the Form tree. Directory Editor will immediately save all changes made to that form.

Creating New Forms

To create a new form, use the following instructions:

1. Click the New Form button (located at the top and bottom of the Forms page).
A new page displays (see the following figure).

Figure 5-13 Creating a New Form

The screenshot shows a window titled "Forms". At the top, there is a "Structural Class:" label followed by a dropdown menu currently showing "account". Below this is a section labeled "Auxiliary Classes:" which contains a list of class names: bootableDevice, certificationAuthority, certificationAuthority-V2, dcObject, extensibleObject, ieee802Device, inetAdmin, inetDomain, and inetSubscriber. To the right of this list is a vertical scrollbar and a set of navigation buttons (up, down, left, right, and a plus sign). At the bottom left of the window is a button labeled "Select Classes".

2. Use the Structural Class menu to specify the primary object class for a directory entry.

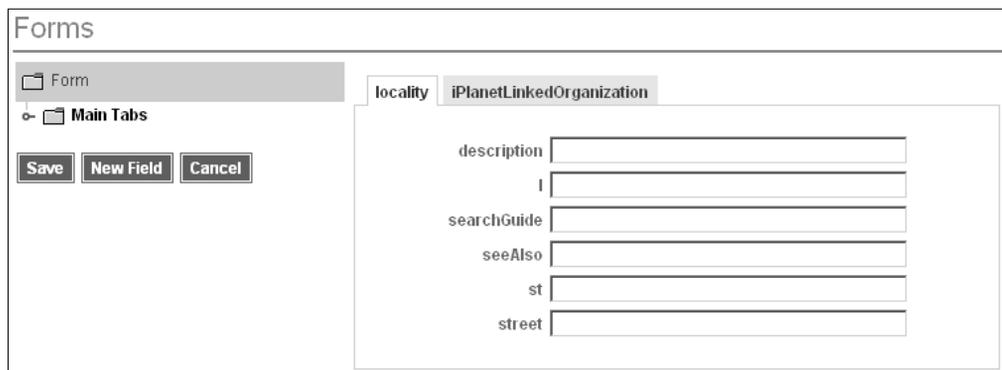
3. Use the **Auxiliary Class** selection tool (if applicable) to select one or more auxiliary classes (additional object classes that allow the entry to have more attributes) for your new form as follows:
 - Select one or more auxiliary classes from the available list and click the  button to move them to the selected list. (Press your Shift key and click on items in the list to select multiple auxiliary classes.)
 - Click the  button to move all available auxiliary classes to the selected list.
 - Click the  button to move all auxiliary classes from the selected list back to the available list.
 - Select auxiliary classes from the selected list and click the  button to move them back to the available list.
 - Use the  (move up) and  (move down) buttons to change the order of auxiliary classes in the selected list
4. After making your selections, click the Select Classes button to add the classes to your new form.

Another page displays, containing a tab for the structural class and one tab for each auxiliary class you selected.

NOTE This page is a preview of the form you are creating.

For example, if you specified the `locality` structural class and the added `iPlanetLinkedOrganization` Auxiliary class, the new page would look like the following figure:

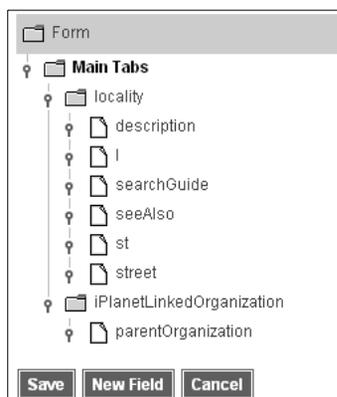
Figure 5-14 Adding Classes to a Form



The tab names, and the content of each tab, will differ with each structural class and auxiliary class selection.

5. Complete the sub-element text boxes on each tab, being certain to provide all required information (identified by a red asterisk).
6. Select the Form, Tab, and field element nodes to specify the field properties (see page 5-79) and to specify the field locations (see page 5-74) for each.

Figure 5-15 Specify Properties for Each Node in the Tree



7. When you are finished, click Save to save your new form to the directory.

Deleting Customized Forms

When you customize a default form, Directory Editor adds a Delete Customized button to the Action column on the main Forms page so you can start over with a customization.

Click this button, and Directory Editor immediately removes the customized form. (You cannot delete any of the Directory Editor default forms.)

Testing Your Form

The best way to test your form before loading it into your production environment, is to preview the form while you are editing it.

In addition, you may want to create a staging or test environment that you can use when you are developing new forms. You can export the staged forms (select Configuration > Import/Export), and then import them into your production environment.

NOTE For importing and exporting instructions, see “Backing Up and Restoring Configurations” on page 161.

Searching Directories

You can use the Directory Editor Search tab to search for objects in the directory installation.

This chapter explains how to configure and execute three different types of searches from the Directory Editor Search page. In addition, this chapter explains how to change the appearance and content of your Search page. The information is organized as follows:

- “Executing a Basic Search” on page 120
- “Executing an Advanced Search” on page 123
- “Executing a Filtered Search” on page 129
- “Editing the Search Page Configuration” on page 131

NOTE If you are *not* using a VLV index, Directory Editor will return a maximum of 100 results. If a search yields more than 100 entries, a note will display above the Results table advising you that the results exceeded the 100-entry limit.

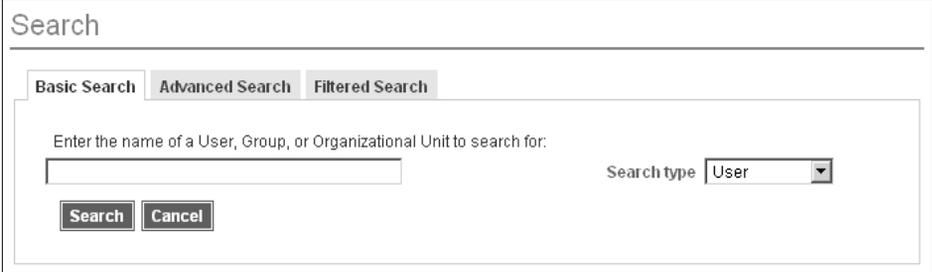
If there are too many entries returned, consider narrowing your search criteria or changing the Number of Results value (up to the limit imposed by Directory Server) on the Search Configuration page (Configure > Search Options).

If you are using a VLV index, this limit does not apply.

Executing a Basic Search

Use the Basic Search tab (the default page) to search for users, groups, or organizational units.

Figure 6-1 Search Results



The screenshot shows a web interface titled "Search". At the top, there are three tabs: "Basic Search", "Advanced Search", and "Filtered Search". The "Basic Search" tab is selected. Below the tabs, there is a text input field with the placeholder text "Enter the name of a User, Group, or Organizational Unit to search for:". To the right of the input field is a dropdown menu labeled "Search type" with "User" selected. Below the input field and dropdown menu are two buttons: "Search" and "Cancel".

To perform a basic search, use the following steps:

1. Enter the name of a user, a group, or an organizational unit into the text field provided.

For example, type **manager** for a list of all objects that have “manager” in their name.

2. Select a Search Type from the drop-down menu.
3. Click the Search button or press Return (Enter) on your keyboard and the search results will display in table-format at the bottom of the Search page.

For example, the following figure shows the result of searching for *manager*.

Figure 6-2 Search Results

Search

Basic Search
Advanced Search
Filtered Search

Enter the name of a User, Group, or Organizational Unit to search for:

Search type All

Search
Cancel

Create
Delete
Rename
Enable
Disable

Results

<input type="checkbox"/>	Name ▼	First Name	Last Name
<input type="checkbox"/>	Accounting Managers		
<input type="checkbox"/>	HR Managers		
<input type="checkbox"/>	QA Managers		
<input type="checkbox"/>	PD Managers		

If search results exceed the maximum number of lines permitted for a single page, Directory Editor will continue the Results table on a new page and display the following navigation tool.



Use this navigation tool to view the different pages as follows:

- Enter a page number into the text box and click the Go button to see a particular page in the set.
- Click to go forward one page or to go back one page.
- Click to go to the first page in the set or click to go to the last page in the set.

NOTE You can change the default setting for maximum number of lines to display per page. See “Configuring the Results Table” on page 134 for more information.

A bank of buttons also displays just above the Results table. Use these buttons in conjunction with the table.

Note that each entry in the Results table is preceded by a checkbox. You can enable one or more checkboxes to select those rows for an action and then click one of the following buttons:

- **Create:** Opens a Create page so you can add a new user, group, or organizational unit. (See Chapter 4, “Creating and Editing Objects.”)
- **Delete:** Opens an Delete page, where you must confirm whether to delete the object. Click Delete to confirm the action or click Cancel to terminate the action.
- **Rename:** Opens a Rename page so you can specify a new name for that object and for children of the object. (See Chapter 4, “Creating and Editing Objects.”)
- **Enable:** Opens an Enable page, where you must confirm whether to enable the object. Click Enable to confirm the action or click Cancel to terminate the action.
- **Disable:** Opens an Disable page, where you must confirm whether to disable the object. Click Disable to confirm the action or click Cancel to terminate the action.

NOTE You can also click any linked entry in the table’s Name column, to open the Edit an Object page where you can edit the object’s attributes. See Chapter 4, “Creating and Editing Objects.” for more information about editing objects.

Executing an Advanced Search

Select the Advanced Search tab to search for objects by specifying search conditions, which can be combined with and operations.

Figure 6-3 Advanced Search Tab

Search

Basic Search | **Advanced Search** | Filtered Search

Search type: User

Base Context: **Browse**

Search Conditions

	Attribute	Condition	Value
<input type="checkbox"/>	audio	contains	

Remove Selected Attribute(s) **Add Attribute**

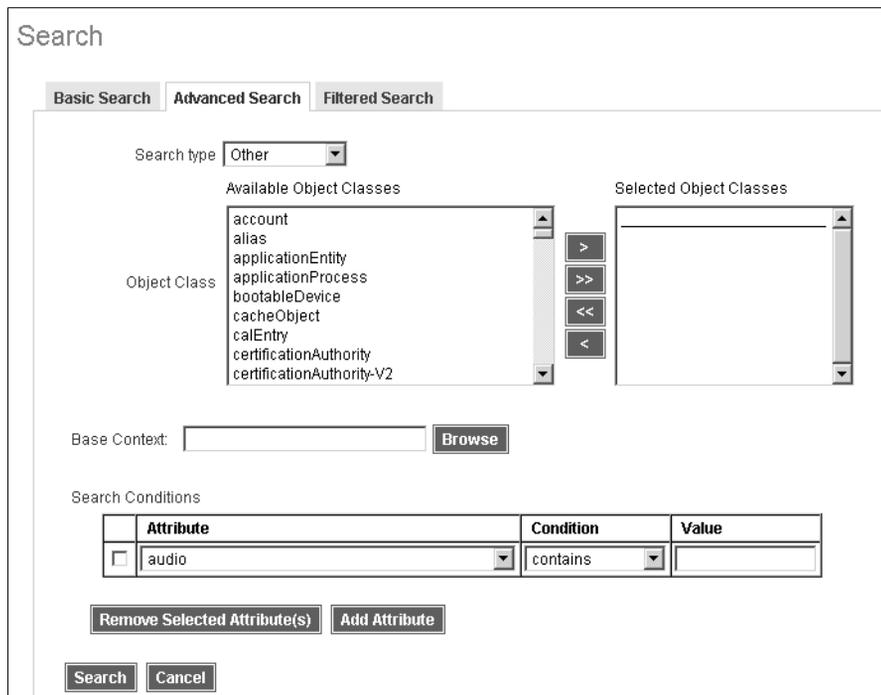
Search **Cancel**

To configure and execute an advanced search, use the following steps:

1. Select one of the following object types from the Search type menu:
 - o **User**
 - o **Group**
 - o **Organizational Unit**
 - o **Other**

When you select the Other option, the Object Classes selection tool displays (similar to the one shown in the following figure).

Figure 6-4 Selecting Object Classes



- You can select object classes from the Available Object Classes list and click the Add button  to move the object classes to the Selected Object Classes list.
 - Click Add All  to move all available object classes to the Selected Object Classes list.
 - Click Remove All  to move all object classes from the Selected Object Classes list back to the Available Object Classes list.
 - Select object classes from the Selected Object Classes list and click Remove  to move the object classes from the Selected list back to the Available Object Classes list.
2. Enter a Base Context in the text box provided (or click Browse to search the organizational units in the directory).

The Search will return results from items in this container and “contained” containers.

3. Use the Search Conditions table to configure your search criteria, as follows:
 - a. Click the drop-down button in the Attribute column to select an attribute from the list.

-
- NOTE**
- The attributes in this list are determined by the object type specified in Step 1.
 - To add rows to the Search Conditions table, click the Add Attribute button.
 - To Remove rows from the Search Conditions table, enable the checkbox in the leftmost column and then click the Remove Selected Attribute(s) button.
 - You cannot remove all rows in this table. If you try to remove the only remaining row, an error will result.
-

- b. Click the drop-down button in the Conditions column and select one of the following conditions from the list:
 - **contains:** Search for attributes that contain the specified attribute value.
 - **equals:** Search for attributes that match the specified attribute value exactly.
 - **does not equal:** Search for attributes that *do not* match the specified attribute value.
4. Enter a value to search for into the Value column of the table.

-
- NOTE** Each attribute, condition, and value builds a subexpression and all the subexpressions are “AND’ed” together to create the search criteria.
- If you do not provide a value, that subexpression will be ignored.
-

5. Click the Search button.

The search results will display in table-format just below the text field.

NOTE If you are *not* using a VLV index, Directory Editor will return a maximum of 100 results. If a search yields more than 100 entries, a note will display above the Results table advising you that the results exceeded the 100-entry limit.

If there are too many entries returned, consider narrowing your search criteria or changing the Number of Results value (up to the limit imposed by Directory Server) on the Search Configuration page (Configure > Search Options).

If you are using a VLV index, this limit does not apply.

The search results will display in table-format at the bottom of the Search page.

As with a Basic Search, if your search results exceed the maximum number of lines permitted for a single page, Directory Editor will continue the Results table on a new page and display a navigation tool. Use this navigation tool to view the different pages as described on page 121

In addition, a bank of buttons (Create, Delete, Rename, Enable, and Disable) also displays just above the Results table. Use these buttons in conjunction with the table as described in Step 2 on page 124.

NOTE Click any of the linked entries in the Name column, and the Edit an Object page displays so you can edit the object’s attributes. (See Chapter 4, “Creating and Editing Objects” for information about editing objects.)

Following are several examples of an Advanced Search.

Example 1: Someone reports a “fender-bender” in the parking lot and you want to notify the car owner. You could specify the following parameters to identify the owner:

- **Search type:** User
- **Base Context:** dc=example,dc=com
- **Search Conditions:**

Table 6-1 Example Search Parameters

Attribute	Condition	Value
carlicense	equals	<license_number>

NOTE Each attribute, condition, and value builds a subexpression, and all the subexpressions are “AND’ed” together to create the search criteria.

If you do not provide a value, that subexpression will be ignored.

The following figure shows the result of searching for the license number:

Figure 6-5 Example 1 Search Results

The screenshot shows an advanced search interface with three tabs: Basic Search, Advanced Search (selected), and Filtered Search. The search type is set to 'User'. The base context is 'dc=example,dc=sun,dc=com'. A search condition is defined with the attribute 'carLicense', condition 'equals', and value 'KF1111'. Below the search criteria are buttons for 'Remove Selected Attribute(s)', 'Add Attribute', 'Search', and 'Cancel'. At the bottom, there are buttons for 'Create', 'Delete', 'Rename', 'Enable', and 'Disable'. The results section shows a table with columns for Name, First Name, and Last Name, containing one entry for 'K. C. Jones'.

Attribute	Condition	Value
carLicense	equals	KF1111

Name	First Name	Last Name
K. C. Jones	K. C.	Jones

Example 2: You want to send an email to all Marketing managers. You could specify the following parameters to identify these managers:

- **Search type:** User
- **Base Context:** dc=example,dc=com
- **Search Conditions:**

Table 6-2 Example Search Parameters

Attribute	Condition	Value
businessCategory	equals	Marketing

Table 6-2 Example Search Parameters

employeeType	equals	Manager
--------------	--------	---------

- NOTE**
- Each attribute, condition, and value builds a subexpression, and all the subexpressions are “AND’ed” together to create the search criteria.
- If you do not provide a value, that subexpression will be ignored.
- Remember to click the Add Attribute button to add rows (additional attributes) to the Search Conditions table.

The following figure shows the result of searching for Marketing managers:

Figure 6-6 Example 2 Search Results

The screenshot shows the Advanced Search interface with the following components:

- Search type: User
- Base Context: dc=example,dc=sun,dc=com
- Search Conditions table:

Attribute	Condition	Value
<input type="checkbox"/> businessCategory	equals	Marketing
<input type="checkbox"/> employeeType	contains	Manager

Buttons: Remove Selected Attribute(s), Add Attribute, Search, Cancel

Buttons: Create, Delete, Rename, Enable, Disable

Results

Name	First Name	Last Name
<input type="checkbox"/> ou=People,dc=central,dc=sun,dc=com		
<input type="checkbox"/> Kenneth Suter	Kenneth	Suter
<input type="checkbox"/> Bob Jones	Bob	Jones

Executing a Filtered Search

Select the Filtered Search tab to specify free-form query strings to search for objects in the directory.

Figure 6-7 Filtered Search tab



Use the default search string or enter your own query in the Search Filter text field.

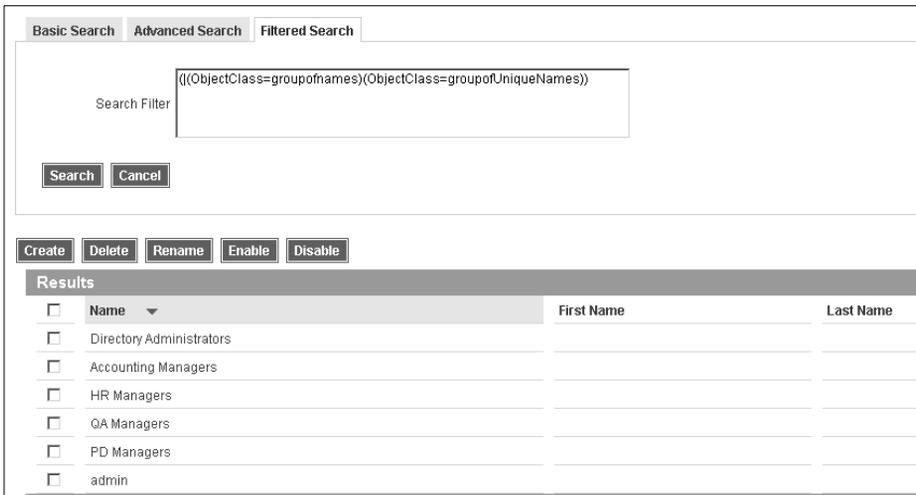
For example, if you want to search for all groups in the directory, type the following text in the Search Filter text field and click Search:

```
( |(ObjectClass=groupofnames)(ObjectClass=groupofUniqueNames) )
```

NOTE You can change the default search string using the Search Configuration page (Configure > Search Options). See “Specifying Default Settings” on page 133 for more information.

The following figure shows the result of the filtered search.

Figure 6-8 Example Filtered Search Results



As with the other search methods, the search results will display in table-format at the bottom of the Search page.

- If you are not using *Virtual List View* (VLV), Directory Editor will return a maximum of 100 entries on up to four pages (25 lines per page by default). If multiple pages are necessary, Directory Editor provides a navigation tool that enables you to view the different pages. (See page 121.)
- If you are using VLV, there is no limit to the number of entries displayed. Directory Editor will provide the navigation tool to enable you to view the different pages.

NOTE

- Directory Editor’s browse and search features both use an LDAP protocol facility called *Virtual List View* (VLV). This facility provides an LDAP client with the ability to specify a “window” into the data that is available on the LDAP server.

For more information about VLV, see Appendix C, “Improving Performance of Browse and Search Features” and the Sun Java™ System Directory Server product documentation.

- If necessary, you can change the default limit for the number of objects displayed by selecting **Configure > Search Options**.
-

In addition, a bank of buttons (Create, Delete, Rename, Enable, and Disable) displays just above the Results table. Use these buttons in conjunction with the table as described in step 2 on page 6-82.

NOTE Click any of the linked entries in the Name column, and the Edit an Object page displays so you can edit the object's attributes. (See Chapter 4, "Creating and Editing Objects" for information about editing objects.)

Editing the Search Page Configuration

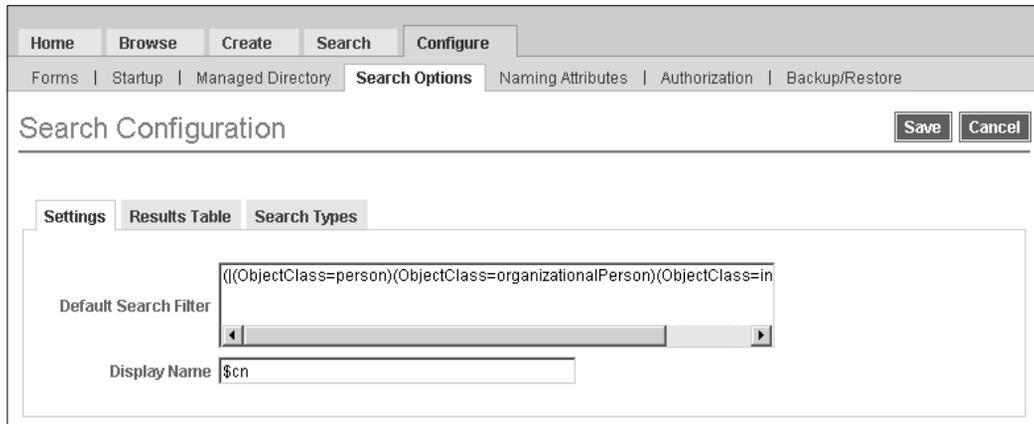
The procedures in this section describe how to change the appearance and content of your Search page. The information is organized as follows:

- "Accessing the Search Configuration Page" on page 132
- "Specifying Default Settings" on page 133
- "Configuring the Results Table" on page 134
- "Specifying Search Types" on page 137

Accessing the Search Configuration Page

To open the Search Configuration page, select the Configure > Search Options. The Search Configuration page is displayed as follows:

Figure 6-9 Search Configuration Page



This page contains three tabs:

- **Settings:** Use this tab to configure a default search filter for the Filtered Search page.
- **Results Table:** Use this tab to configure a results table for the Search page.
- **Search Types:** Use this tab to specify the type of searches available for the Search page.

Specifying Default Settings

Use the following procedure to configure a default search filter for the Search page.

1. Select the Settings tab (see Figure 6-9).
2. Modify the default filter shown in the Default Search Filter text box or enter a new filter.

CAUTION Improper use of a search filter can impact both Directory Editor and Directory Server performance. For detailed information about constructing filters, refer to the *Sun Java™ System Directory Server 5 2004Q2 Administration Guide*.

3. Modify or enter a new display name in the Display Name text box.

You may want Directory Editor to return results in the same way your company displays data. For example; `cn`, `uid`, and so forth.

NOTE The dollar sign (\$) that precedes this value denotes the start of an attribute name.

4. You can click Save to apply your changes (or click Cancel and return to the Home page).

NOTE You do not have to Save your changes on each page — you can wait until you are finished working with all of the tabs and then save your changes to the directory.

From here, you can select another tab to continue configuring your Search page.

Configuring the Results Table

Use the following procedure to configure a results table for the Search page.

1. Select the Results Table tab.

Figure 6-10 Results Table Page

The screenshot shows the configuration interface for the Results Table. It is organized into two main sections: **Results Columns** and **Favored Attributes**.

Results Columns Section:

- Available Attributes:** A list containing abstract, accountUnlockTime, aci, aclRights, aclRightsInfo, adminRole, adminUrl, administratorContactInfo, and aliasedObjectName.
- Selected Attributes:** A list containing givenname and sn.
- Display Names Table:**

Attribute Name	Display Name
dn	Name
givenname	First Name
sn	Last Name
- Column Width:** 25
- Maximum Number of results without VLV:** 100
- Number of results per page:** 25

Favored Attributes Section:

- Search Types for Favored Attributes:** User
- Available Attributes:** A list containing audio, businessCategory, carLicense, departmentNumber, description, destinationIndicator, displayName, employeeNumber, and employeeType.
- Selected Attributes:** A list containing uid, cn, givenName, and sn.

2. Use the Results Columns selection tool to add attributes to or remove attributes from the Selected Attributes list, as follows:
 - Select one or more attributes from the Available Attributes list and click the  button to move them to the Selected Attributes list. (Press your Shift key and click on items in the list to select multiple attributes.)
 - Click the  button to move all available attributes to the Selected Attributes list.
 - Click the  button to move all attributes from the Selected Attributes list back to the Available Attributes list.
 - Select one or more attributes from the Selected Attributes list and click the  button to move them back to the Available Attributes list.
 - Use the  (move up) and  (move down) buttons to change the order of attributes in the Selected Attributes list.
3. By default, Directory Editor populates the Display Names table with Attribute Names and Display Names.

You can replace the default names in the Display Names column by typing directly into the text boxes provided.

4. Enter a value in the Column Width field to specify a maximum width (maximum character length) for columns in the Results table.
5. Enter a value in the Maximum Number of results without VLV field to specify the maximum number of search results (how many lines) to display in the Results table if you are not using the Virtual List View LDAP protocol facility.

-
- NOTE**
- Directory Editor's browse and search features both use an LDAP protocol facility called *Virtual List View* (VLV). This facility provides an LDAP client with the ability to specify a "window" into the data that is available on the LDAP server.

For more information about VLV, see Appendix C, "Improving Performance of Browse and Search Features" and the Sun Java™ System Directory Server product documentation.
 - Changing the Maximum Number of results without VLV value will affect *both* a standard search and the group viewing features.
-

6. Enter a value in the Number of results per page field to specify the maximum number of search results (how many lines) to display in the Results table on a single page.

If search results exceed this maximum number of lines, Directory Editor will continue the Results table on a new page. Use the navigation tool to view the different pages as follows:



- Enter a page number into the text box and click the Go button to see a particular page in the set.
 - Click  to go forward one page or  to go back one page.
 - Click  to go to the first page in the set or click  to go to the last page in the set.
7. Use the Search Types for Favored Attributes menu to select an attribute type (User, Group, or Organization) for the favored attributes to be selected in Step 8.
 8. Use the Favored Attributes selection tool to specify which attributes are available (and the order in which they display) when you click the Add Attributes button on the Advanced Search page:
 - Select one or more attributes from the Available Attributes list and click the  button to move them to the Selected Attributes list. (Press your Shift key and click on items in the list to select multiple attributes.)
 - Click the  button to move all available attributes to the Selected Attributes list.
 - Click the  button to move all attributes from the Selected Attributes list back to the Available Attributes list.
 - Select one or more attributes from the Selected Attributes list and click the  button to move them back to the Available Attributes list.
 - Use the  (move up) and  (move down) buttons to change the order of attributes in the Selected Attributes list.

Specifying Search Types

You can use the Search Types page to modify the kind of search Directory Editor performs by default.

Use the following procedure to specify the type of searches available for the Search page.

1. Select the Search Types tab.

Figure 6-11 Search Types Page

Name	Value				
<input type="checkbox"/> User	<table border="1"> <thead> <tr> <th>Available Objectclasses</th> <th>Selected Objectclasses</th> </tr> </thead> <tbody> <tr> <td>account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2</td> <td>person organizationalPerson inetOrgPerson</td> </tr> </tbody> </table>	Available Objectclasses	Selected Objectclasses	account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2	person organizationalPerson inetOrgPerson
Available Objectclasses	Selected Objectclasses				
account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2	person organizationalPerson inetOrgPerson				
<input type="checkbox"/> Organization	<table border="1"> <thead> <tr> <th>Available Objectclasses</th> <th>Selected Objectclasses</th> </tr> </thead> <tbody> <tr> <td>account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2</td> <td>organizationalUnit organization</td> </tr> </tbody> </table>	Available Objectclasses	Selected Objectclasses	account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2	organizationalUnit organization
Available Objectclasses	Selected Objectclasses				
account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2	organizationalUnit organization				
<input type="checkbox"/> Group	<table border="1"> <thead> <tr> <th>Available Objectclasses</th> <th>Selected Objectclasses</th> </tr> </thead> <tbody> <tr> <td>account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2</td> <td>groupOfNames groupOfUniqueNames</td> </tr> </tbody> </table>	Available Objectclasses	Selected Objectclasses	account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2	groupOfNames groupOfUniqueNames
Available Objectclasses	Selected Objectclasses				
account alias applicationEntity applicationProcess bootableDevice cacheObject calEntry certificationAuthority certificationAuthority-V2	groupOfNames groupOfUniqueNames				

Remove Selected Types(s) Add Type

2. If necessary, change the Default Search Type name by entering a new value in the text box provided.

3. Use the table to change the search type names and associated attributes that are available in the Search Type menu list.

- a. To change the name of a search type, delete the old name and enter a new value.

For example, you might want to change Organization to Offices.

- b. Use the Objectclasses selection tools in the Value column to create a list of valid values for that search type.

For example, you might want to specify the following values for the User type:

- Person
- organizationalPerson
- inetorgPerson (*default*)
- engineeringPerson

NOTE If you create a custom user class, you should add that object class to the user object.

4. When you are finished, click the Save button to save your changes and return to the Home page (or click Cancel to undo your changes).

NOTE You do not have to Save your changes on each page — you can wait until you are finished working with all of the tabs and then save your changes to the directory.

Adding Search Types

To add search types to the table, use the following procedure:

1. Click the Add Type button located beneath the table.
2. When Directory Editor adds a new row to the table, enter a name for the search type in the text box provided in the Name column.

3. Use the Objectclasses selection tool in the Value column to create a list of valid values for the new search type.

For example, you might want to specify a Location search type with the following object classes:

- country
 - friendlyCountry
 - room
4. When you are finished, click Save to save your changes and return to the Home page (or click Cancel to undo your changes).

Removing Search Types

To remove objects from the Search Types menu list,

1. Enable the checkbox next to the search type you want to delete.
2. Click the Remove Selected Types button located beneath the table.
Directory Editor immediately removes that row from the table.
3. Click Save to save your changes and return to the Home page, or click Cancel to undo your changes and return to the Home page.

Configuring Directory Editor

Use the information provided in this chapter to configure your Directory Editor application to control user access to applications and application components, and to define relative distinguished names (RDN) configurations. This chapter is organized as follows:

- “Controlling User Access” on page 141
- “Working with Roles” on page 144
- “Working with Naming Attributes” on page 151
- “Editing the Startup Properties” on page 157
- “Editing the Managed Directory Properties” on page 157

Controlling User Access

Authentication and *Authorization* are terms used to describe methodologies for controlling access to applications or application components.

- **Authentication** is the process by which an application challenges a client to supply credentials — typically providing a user name and password through a log-in page. Based on these credentials, the application determines whether a user (or a different client, such as another application) can use the application.
- **Authorization** assumes that a client has already been authenticated and is in the process of determining whether a client can use a *component* of the application. Authorization takes a more fine-grained approach to access control.

You can use the Directory Editor Authorization page to authenticate clients *and* to control access to Directory Editor components using fine-grained authorization.

NOTE By default, you must have the Manager role to access the Authorization page. For more information about roles, see “Understanding Roles” on page 142.

To understand authorization, you must understand the terms *role*, *principal*, and *capabilities*. These terms are described in the following sections:

- “Understanding Roles” on page 142
- “Understanding Principals” on page 143
- “Understanding Capabilities” on page 143

Understanding Roles

A *role* describes a user's function within the enterprise hosting Directory Editor and determines with which parts of Directory Editor the user can interact.

By default, Directory Editor is pre-configured with two roles:

- **Manager:** This role has access to all of Directory Editor's features and functionality.
- **Default:** This role has more limited capabilities within the application. For example, you can browse directory data but you cannot edit that data.

Directory Editor enables you to add roles that support interactions appropriate for your enterprise, and these roles can consist of individual users or a group of users.

For example, if you create CEO, help desk administrator, and HR administrator roles for your organization, it is probably not necessary for each of these roles to have the same access capabilities.

Every role is associated with a set of *principals* that assume the role (see the next section, “Understanding Principals”).

For Directory Editor, there is a single group in the directory server (called the *Manager Group*) that serves as the principal corresponding to the Manager role. You use the Managed Directory page to specify the Manager Group at configuration time (also available after configuration by selecting Configure > Managed Directory). So, if you have a particular user that should have full access to all Directory Editor functions, make that user's DN a member of the Manager Group.

Understanding Principals

A *principal* represents an entity (such as an individual, corporation, or login ID). The term *subject* is used to describe entities (typically human users). Subjects can be represented by multiple, differing principals — just as people can be represented by their credit card number to banks and by their UNIX account name to system administrators. The credit card numbers and UNIX account names are principals in this case.

Because Directory Editor is focused on directory data management, its principals are all represented using the following directory objects:

- DN (distinguished names)
- Directory Server groups
- Directory Server roles

A user entering a DN in the log-in page can be represented by several different principals, depending on the data in the directory. For example, if the user's account ID happens to be a member of a specific group, that user will be represented by the account ID's DN and by the DN of the group to which the DN belongs.

In Directory Editor, objects representing the user (or subject) are stored in the HTTP session. After the user enters an account ID and a password on the Directory Editor's log-in page, Directory Editor populates the subject with all of the various principals (person entries, groups, and roles) associated with that account ID.

NOTE If you edit the Manager role's default principals, you can restore the original settings by clicking the Restore Default Setting button located on the Principals tab.

Understanding Capabilities

Capabilities are rights to perform actions within Directory Editor. A capability aggregates a set of resources that are necessary to perform the associated action (see Appendix B, "Resources for Capability Configuration").

By default, the Directory Editor capabilities include:

- **Browse:** Enables the user to browse the directory.
- **Configure:** Enables the user to configure the application.

- **Debug:** Enables the user to debug the application.
- **Edit/Create:** Enables the user to edit and create directory entries.
- **Search:** Enables the user to search the directory.

NOTE If you edit the Default role's default capabilities, you can restore the original settings by clicking the Restore Default Setting button located on the Capabilities tab.

Working with Roles

This section provides instructions for defining, editing, and deleting roles. The section is organized as follows:

- “Accessing the Authorization Page” on page 144
- “Creating Directory Editor Roles” on page 145
- “Editing Roles” on page 149
- “Deleting Roles” on page 151

Accessing the Authorization Page

To open the Authorization page, select the Configure tab and then select the Authorization tab. The Authorization page is displayed as follows:

Figure 7-1 Authorization Page

Authorization

Here you can manage roles which control access to views and tasks based on the user's credentials.

Create Role
Edit Selected Role
Delete Selected Role(s)

	Role	Principals	Capabilities
<input type="checkbox"/>	Manager	cn=directory manager cn=directory administrators,dc=central,dc=sun,dc=com	Browse Configure Debug Edit/Create Search
<input type="checkbox"/>	Default	*	Browse Edit/Create Search

This page consists of the following features:

- **Create Role button:** Click this button to create additional roles appropriate for your enterprise. (See “Creating Directory Editor Roles” on page 145.)
- **Edit Selected Role button:** Click this button to change the properties, principals, and capabilities associated with a particular role. (See “Editing Roles” on page 149.)
- **Delete Selected Role(s) button:** Click this button to remove roles. (See “Deleting Roles” on page 151.)
- **Role table:** By default, this table contains the Manager and Default roles.

Creating Directory Editor Roles

Before you can define new Directory Editor roles for your enterprise, you must decide which tasks a common set of users must perform. For example, all of your help desk administrators must have write access to directory data.

After you have identified these tasks, use the following steps to create a new role:

1. Select Configure > Authorization.
2. When the Authorization page is displayed (see Figure 7-1 on page 144), click the Create Role button.
3. On the New Role page, enter a meaningful name into the Role Name text box. For example, Site Managers.

Figure 7-2 Role Properties Tab

The screenshot shows a 'New Role' dialog box. At the top, the title bar reads 'New Role' with 'Save' and 'Cancel' buttons on the right. Below the title bar, there are three tabs: 'Role Properties' (which is selected and highlighted), 'Principals', and 'Capabilities'. The 'Role Properties' tab contains a text input field labeled 'Role Name:'. At the bottom of the dialog, there are 'Save' and 'Cancel' buttons.

- To specify a set of principals for this role, select the Principals tab and then click the Search for Principals button.

Figure 7-3 Principals Tab

- When the Search for Principals page is displayed (Figure 7-4), use one of the search tabs (Basic, Advanced, or Filtered) to search the directory for identity, group, or role objects you want to assume the new role.

Figure 7-4 Search for Principals Page

- Define the parameters for your search and then click the Search button.

NOTE If necessary, see Chapter 6, “Searching Directories” to review the instructions for using these search tabs.

For example, if you want all Managers to assume the new role, you can use the Basic Search tab to search for Directory Administrators as follows:

Figure 7-5 Adding Objects to the Principals Set

Search for Principals

Search the directory for identity, group, and role objects to be added to the set of principals that assume the edited role.

Basic Search | Advanced Search | Filtered Search

Enter the name of a User, Group, or Organizational Unit to search for:

Managers Search type: Group

Search Cancel

Add Selected Principals

Results From Search of : dc=central,dc=sun,dc=com Page: 1 of 1 Go

<input type="checkbox"/>	Name	First Name	Last Name
<input checked="" type="checkbox"/>	Accounting Managers		
<input checked="" type="checkbox"/>	HR Managers		
<input checked="" type="checkbox"/>	PD Managers		
<input checked="" type="checkbox"/>	QA Managers		

Page: 1 of 1 Go

- In the Results from Search table, enable the Results checkbox(es) to select principals for the new role and then click the Add Selected Principals button.

The New Role page redisplayes and the Principal table now contains the principal(s) you specified.

Figure 7-6 Principal Table

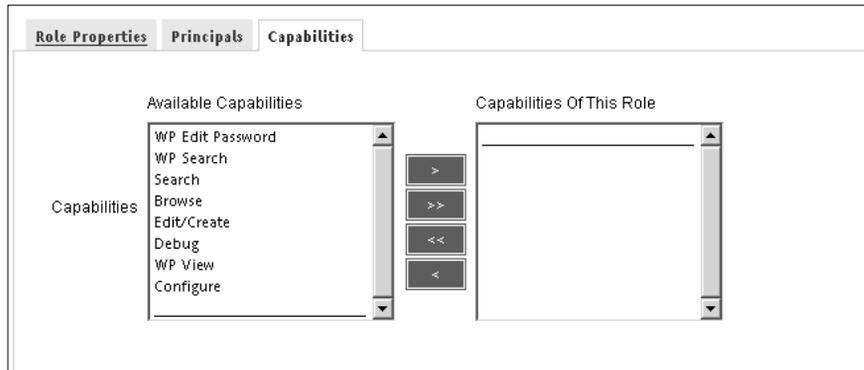
<input type="checkbox"/>	Principal
<input type="checkbox"/>	cn=Accounting Managers,ou=groups,dc=central,dc=sun,dc=com
<input type="checkbox"/>	cn=HR Managers,ou=groups,dc=central,dc=sun,dc=com
<input type="checkbox"/>	cn=PD Managers,ou=groups,dc=central,dc=sun,dc=com
<input type="checkbox"/>	cn=QA Managers,ou=groups,dc=central,dc=sun,dc=com

Search for Principals... Remove Selected

NOTE To remove principals from the Principal table (and from the new role), enable the checkbox to the right of the principal name(s), and then click the Remove Selected button.

8. Select the Capabilities tab (Figure 7-7) to specify a set of actions that can be performed by users who assume the new role.

Figure 7-7 Capabilities Tab



- Select one or more capabilities from the Available Capabilities list and click  to move them to the Capabilities Of This Role list. (Press your Shift key and click on items in the list to select multiple resources.)
- Click  to move all resources to the Capabilities Of This Role list.
- Click  to move all resources from the Capabilities Of This Role list back to the Available Capabilities list.
- Select resources from the Capabilities Of This Role list and click  to move them back to the Available Capabilities list.

For example, you might want to assign all of the capabilities to the Help Desk Administrator role.

9. Click Save to save the new role and to add it to the Roles table (or click Cancel to return to the Authorization page without saving your changes).

Figure 7-8 shows the updated Roles table.

Figure 7-8 New Role Added to the Roles Table

Authorization			
Here you can manage roles which control access to views and tasks based on the user's credentials.			
<input type="button" value="Create Role"/> <input type="button" value="Edit Selected Role"/> <input type="button" value="Delete Selected Role(s)"/>			
	Role	Principals	Capabilities
<input type="checkbox"/>	Site Managers	cn=hr managers,ou=groups,dc=central,dc=sun,dc=com cn=pd managers,ou=groups,dc=central,dc=sun,dc=com cn=accounting managers,ou=groups,dc=central,dc=sun,dc=com cn=qa managers,ou=groups,dc=central,dc=sun,dc=com	Browse Configure Edit/Create Search WP Search WP View
<input type="checkbox"/>	Help Desk Administrator	cn=Directory Administrators,dc=central,dc=sun,dc=com uid=KC,ou=People,dc=central,dc=sun,dc=com	Browse Configure Debug Edit/Create Search WP Edit Password WP Search
<input type="checkbox"/>	Manager	cn=contractors,ou=groups,dc=central,dc=sun,dc=com cn=directory administrators,dc=central,dc=sun,dc=com cn=pd managers,ou=groups,dc=central,dc=sun,dc=com cn=subcontractors,ou=groups,dc=central,dc=sun,dc=com cn=engineers,ou=groups,dc=central,dc=sun,dc=com	Browse Configure Debug Edit/Create Search WP Edit Password WP Search WP View
<input type="checkbox"/>	Default	*	Browse Edit/Create Search WP Search

Editing Roles

To edit selected authorization roles, use the following steps:

1. Select Configure > Authorization.
2. When the Authorization page is displayed, click the checkbox located next to the role you want to edit.

Figure 7-9 Click the Checkbox

<input checked="" type="checkbox"/>	Help Desk Administrators
-------------------------------------	--------------------------

3. Click the Edit Selected Role button to open the Edit page.

Figure 7-10 Edit Page



The screenshot shows a web application window titled "Help Desk Administrator - Edit". In the top right corner of the window, there are two buttons: "Save" and "Cancel". Below the title bar, there are three tabs: "Role Properties", "Principals", and "Capabilities". The "Role Properties" tab is currently selected. Inside this tab, there is a text input field labeled "Role Name:" which contains the text "Help Desk Administrator".

4. The process for editing a role is the same as the process you used to create it. Review the instructions provided in “Creating Directory Editor Roles” on page 145 if necessary.

NOTE

- If you edit the Manager role’s default principals, you can restore the original settings by clicking the Restore Default Setting button located on the Principals tab.
 - Editing the Manager principals on the Principals tab is the same as editing the Manager Principal fields on the Managed Directory page (Configure > Managed Directory).
 - If you edit the Default role’s default capabilities, you can restore the original settings by clicking the Restore Default Setting button located on the Capabilities tab.
-

5. When you are finished, click Save (or click Cancel to return to the Authorization page without saving your changes).

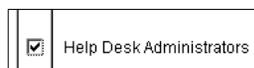
Deleting Roles

NOTE You cannot delete the Manager or the Default roles.

To delete selected roles, use the following steps:

1. Select Configure > Authorization.
2. When the Authorization page is displayed, enable the checkbox(es) located next to the role(s) you want to delete.

Figure 7-11 Click the Checkbox



3. Click the Delete Selected Role(s) button and Directory Editor will immediately remove the selected role(s) from the Roles table.

Working with Naming Attributes

This section provides instructions for defining, editing, and deleting naming attributes. The section is organized as follows:

- “Accessing the Naming Attributes Page” on page 152
- “Creating New Object Class - Naming Attribute Mappings” on page 153
- “Editing Naming Attributes” on page 155
- “Deleting Selected Naming Attributes” on page 156

Accessing the Naming Attributes Page

To create new objects, Directory Editor must know how to construct DNs (distinguished names) for the new objects.

For example, if your customer wants to use `uid` (user ID) as the naming attribute for `inetOrgPerson` instead of `cn` you might specify the following DN for a newly created entry:

```
cn=Mike Miller,dc=example,dc=com
```

instead of:

```
uid=mmiller,dc=example,dc=com
```

Directory Editor ships with a small set of default naming attributes to use for object classes, so it is important that you modify these mappings to match the naming conventions used by your enterprise for naming directory objects. You must configure any object class that you add to the Create page with naming attributes.

To access the Naming Attributes page,

1. Select the Configuration > Naming Attributes.

The Naming Attributes page is displayed as follows:

Figure 7-12 Naming Attributes Page

	Object Class	Naming Attributes
<input type="checkbox"/>	mailgroupmember	mgmemMemberOfGroup
<input type="checkbox"/>	groupofnames	cn
<input type="checkbox"/>	locality	l
<input type="checkbox"/>	domain	dc
<input type="checkbox"/>	person	cn
<input type="checkbox"/>	mailgroup	cn
<input type="checkbox"/>	inetorgperson	uid
<input type="checkbox"/>	groupofuniquenames	cn
<input type="checkbox"/>	organizationalunit	ou
<input type="checkbox"/>	organization	o

This page consists of the following features:

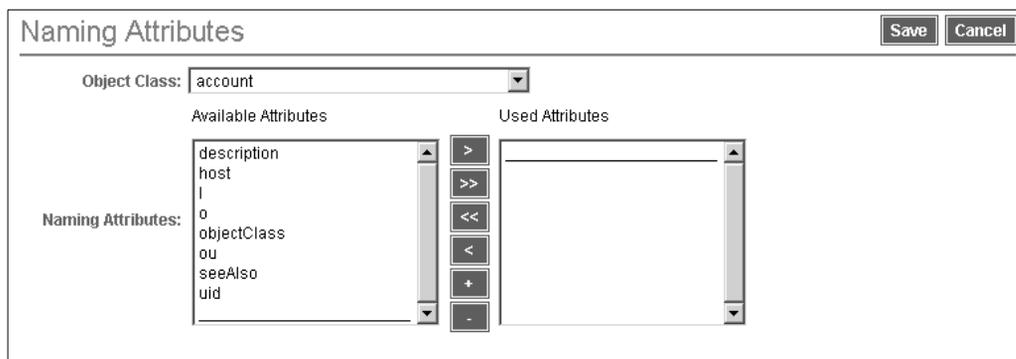
- **Naming Attribute Mapping table:** By default, this table contains a small set of object classes and their naming attributes.
- **New button:** Click this button to add object classes (and define naming attributes for those object classes) to your enterprise. (See “Creating New Object Class - Naming Attribute Mappings” on page 153.)
- **Edit Selected button:** Click this button to edit the naming attributes currently selected for the selected object class. (See “Editing Naming Attributes” on page 155.)
- **Delete Selected button:** Click this button to remove an object class and associated naming attributes from the table. (See “Deleting Selected Naming Attributes” on page 156.)

Creating New Object Class - Naming Attribute Mappings

After you have identified these tasks, use the following steps to create a new naming attribute:

2. Select Configure > Naming Attributes.
3. When the Naming Attributes page is displayed (see Figure 7-12 on page 152), click the New button.
4. A new Naming Attributes page is displayed (Figure 7-13). Select the object class from the Object Class menu.

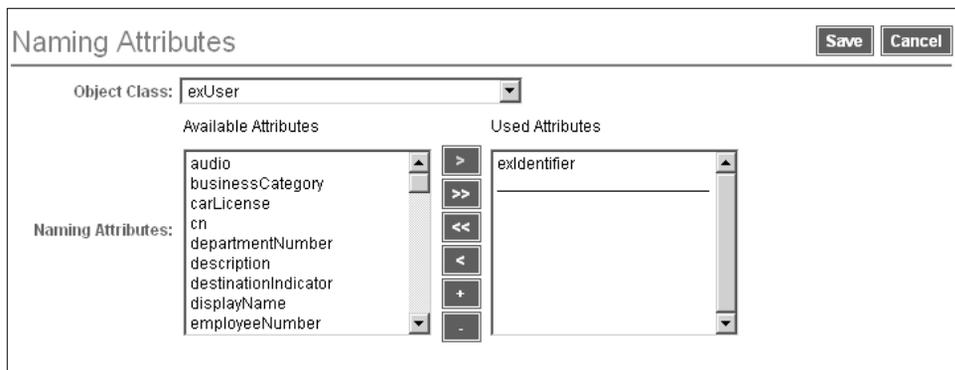
Figure 7-13 New Naming Attributes Page



5. Use the Naming Attributes selection tool to specify naming attributes for the new object class, as follows:
 - Select one or more naming attributes from the Available Attributes list and click the  button to move them to the Used Attributes list. (Press your Shift key and click on items in the list to select multiple naming attributes.)
 - Click the  button to move all naming attributes to the Used Attributes list.
 - Click the  button to move all naming attributes from the Used Attributes list back to the Available Attributes list.
 - Select naming attributes from the Used Attributes list and click the  button to move them back to the Available Attributes list.
 - Use the  (move up) and  (move down) buttons to change the order of attributes in the Used Attributes list

For example, you might specify a new objectclass called `exUser` for extending the default user object and have an attribute called `exIdentifier` as the naming attribute.

Figure 7-14 New Object Class and Naming Attribute Added to the Table



- Click Save to save the new object class and attribute(s) (or click Cancel to return to the Naming Attributes page without saving your changes).

Figure 7-15 shows the new entry added to the Object Class table.

Figure 7-15 Updated Table

	Object Class	Naming Attributes
<input type="checkbox"/>	groupofnames	cn
<input type="checkbox"/>	exuser	exidentifier
<input type="checkbox"/>	locality	l
<input type="checkbox"/>	domain	dc
<input type="checkbox"/>	person	cn
<input type="checkbox"/>	mailgroup	cn
<input type="checkbox"/>	inetorgperson	uid
<input type="checkbox"/>	groupofuniquenames	cn
<input type="checkbox"/>	organizationalunit	ou
<input type="checkbox"/>	organization	o

Editing Naming Attributes

To edit selected naming attributes, use the following steps:

- Select Configure > Naming Attributes.
- When the Naming Attributes page is displayed, click the checkbox located next to the role you want to edit.
- Click the Edit Selected button to open a new Naming Attributes page (similar to Figure 7-16).

Figure 7-16 Editing the Naming Attributes

Naming Attributes

Object Class: person

Available Attributes

- description
- objectClass
- seeAlso
- sn
- telephoneNumber
- userPassword

Used Attributes

- cn

Naming Attributes:

Note that the Object Class menu is not available on this page. Instead, Directory Editor displays the selected object class name.

4. Use the Naming Attributes selection tool to add or remove naming attributes. Review the instructions provided in “Creating New Object Class - Naming Attribute Mappings” on page 153 if necessary.
5. When you are finished, click Save (or click Cancel to return to the Authorization page without saving your changes).

Deleting Selected Naming Attributes

To delete selected naming attributes, use the following steps:

1. Select Configure > Naming Attributes.
2. When the Naming Attributes page is displayed, enable the checkbox(es) located next to the object class(es) you want to delete.

Figure 7-17 Click the Checkbox

	Object Class	Naming Attributes
<input type="checkbox"/>	groupofnames	cn
<input type="checkbox"/>	locality	l
<input type="checkbox"/>	domain	dc
<input type="checkbox"/>	person	cn
<input checked="" type="checkbox"/>	mailgroup	mgrpDeliverTo,owner
<input type="checkbox"/>	inetorgperson	cn
<input type="checkbox"/>	groupofuniquenames	cn
<input type="checkbox"/>	organizationalunit	ou
<input type="checkbox"/>	organization	o

3. Click the Delete Selected button and Directory Editor will immediately remove the selected object class(es) from the table.

Editing the Startup Properties

After initially configuring the Startup Properties page, you can edit any of the property values by selecting Configuration > Startup.

The steps for editing any of the properties provided on this tab are the same as the steps you performed during the initial configuration.

Editing the Managed Directory Properties

After initially configuring the Managed Directory page, you can edit any of the specified property values by selecting Configuration > Managed Directory.

The steps for editing any of the properties provided on this tab are the same as the steps you performed during the initial configuration — except for the Manager Principals parameter.

After completing the initial configuration of Directory Editor, the program adds a Search for Principals button beneath the Manager Principals text boxes so you can search the directory for principals.

NOTE Completing these Manager Principals fields is the same as selecting Configure > Authorization and editing the Manager role's principals on the Principals tab.

To search for principals to add to the Manager Principals set,

1. Click the Search for Principals button.

Figure 7-18 Search for Principals button



- When the Search for Principals page is displayed (Figure 7-19), select one of the search tabs (Basic, Advanced, or Filtered) to search the directory for User, Organization, Group, or All objects.

Figure 7-19 Search for Principals Page

- Define the parameters for your search and then click the Search button.

NOTE If necessary, see Chapter 6, “Searching Directories” to review the instructions for using these search tabs.

For example, if you add all Directory Administrators to the Manager Principals list, you can select the Basic Search tab to search for Directory Administrators as follows:

Figure 7-20 Adding Objects to the Principals Set

- When a Results table is displayed with the results of your search, enable the appropriate Results checkbox(es) to select those principals and then click the Add Selected Principals button.

The Managed Directory page redisplay. Note that the Manager Principals list now contains the principal(s) you specified.

Figure 7-21 New Manager Principals List

* Manager Principals:	<input type="text" value="cn=contractors,ou=groups,dc=central,dc=sun,dc="/> ⊕ Add ⊖ Rem
	<input type="text" value="cn=directory administrators,dc=central,dc=sun,dc="/> ⊕ Add ⊖ Rem
	<input type="text" value="cn=pd managers,ou=groups,dc=central,dc=sun,dc="/> ⊕ Add ⊖ Rem
	<input type="text" value="cn=subcontractors,ou=groups,dc=central,dc=sun,dc="/> ⊕ Add ⊖ Rem
	<input type="text" value="cn=engineers,ou=groups,dc=central,dc=sun,dc="/> ⊕ Add ⊖ Rem
	<input type="text" value="cn=Directory Administrators,dc=central,dc=sun,dc="/> ⊕ Add ⊖ Rem

Backing Up and Restoring Configurations

This chapter describes how to back up your configurations to a file and how to restore configurations from a file using Directory Editor. The information is divided into the following sections:

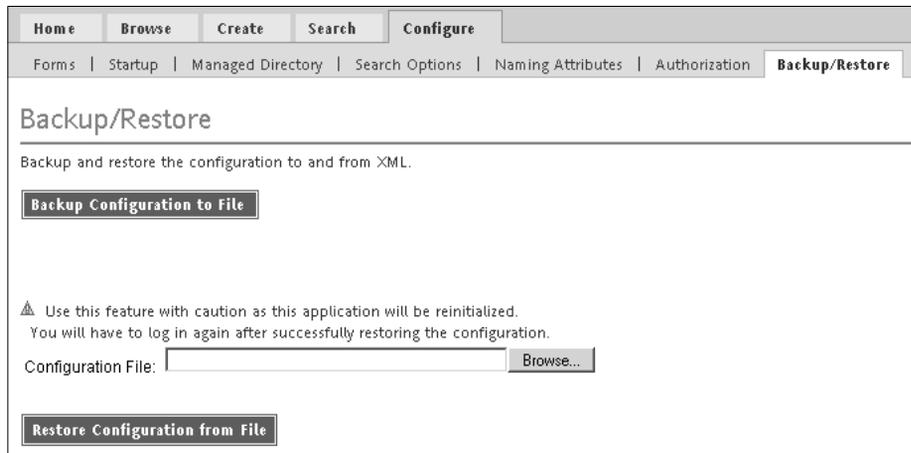
- “Accessing the Backup/Restore Page” on page 162
- “Backing Up Your Configuration” on page 162
- “Restoring Your Configuration” on page 163

CAUTION You must use caution when backing up or restoring configurations. Directory Editor will re-initialize the application, and you will be required to log in again after a successful back up or restore operation.

Accessing the Backup/Restore Page

To open the Backup/Restore page, select the Configure tab and then select the Backup/Restore tab. The Backup/Restore page is displayed as follows:

Figure 8-1 Backup/Restore Page



Backing Up Your Configuration

The Directory Editor back-up feature enables you to save your current configuration to a file or edit your configuration file with an XML editing tool.

To back up your configuration, use the following procedure:

1. Click the Export a Configuration to File button.
2. When the Opening config.xml dialog box displays (Figure 8-2), save the file to disk. (You must specify a file name and specify a folder in which to save the file.)

Figure 8-2 Specifying How Your Browser Should Open the File

3. After making your selection, click OK to close the Opening config.xml dialog box.

You should now be able to open the configuration XML file in your browser. At this point, you can edit the file, and then import it using the instructions provided in the next section, or store the file for backup purposes.

Restoring Your Configuration

To restore a configuration from a file, use the following procedure:

1. Type the configuration path and file name in the Configuration File text box, or click Browse to locate the file you want to restore.

This file should be an XML file you created using the Backup function (described in the previous section). You can import the entire configuration or just part of the configuration as long as the XML is well-formed and valid.

2. Click the Import a Configuration to File button.
 - o If the import is successful the Log-In page is displayed and you must log in again.
 - o If there are errors in your file, Directory Editor will prompt you to correct those errors before trying to import the file again.

Removing the Software

Use the information provided in this chapter to remove Directory Editor from your Windows or UNIX system and how to undeploy Directory Editor from your application server. The information is organized as follows:

- “Removing the Software” on page 165
- “Undeploying Directory Editor from Your Application Server” on page 168

Removing the Software

Use the instructions in one of the following sections to remove the Directory Editor software from your machine

- “Using the Directory Editor Uninstaller” on page 165
- “Uninstalling Manually From a Windows Machine” on page 166
- “Uninstalling Manually From a Unix Machine” on page 167

Using the Directory Editor Uninstaller

Use the following steps to uninstall the software using the Directory Editor uninstaller:

1. Open a command window, and go to the directory where you installed Directory Editor.
2. Type the following command to launch the Uninstall wizard:

```
java uninstall_Directory_Editor
```

3. The Welcome screen displays. Read the information provided, and then click Next.
4. When the Ready to Uninstall screen is displayed, click Uninstall Now.
When the uninstallation is complete, the Uninstallation Summary screen displays to report status.
5. You can click the Details button for more information, or click Close to exit the Uninstall Wizard.
6. To remove any links or references to the Directory Editor software, you must undeploy Directory Editor from your application server. Use the instructions provided in “Undeploying Directory Editor from Your Application Server” on page 168.

Uninstalling Manually From a Windows Machine

NOTE Use this procedure to uninstall Directory Editor *only* if the Directory Editor uninstaller is not operational.

Use the following steps to remove Directory Editor from a Windows machine:

1. Stop your application server.
2. Locate and open the Directory Editor `webapps` directory in your application server's directory.
3. Delete the `de.war` file and the `de` directory.
4. Remove the following `<compid>` elements and sub-elements from the `productregistry` file located in the `C:\WINNT\system32` directory:
`<compid>DE` files
`<compid>Directory Editor war` file
`<compid>Directory Editor`
5. If necessary, return your application server's `\bin` directory and double-click `startup.bat` to restart the application.

6. Remove all Directory Editor configuration information from the Sun Java System Directory Server or Sun ONE Directory Server, which (by default) is located under `Services:DML`.
 - a. Log into Directory Server.
 - b. Select the Directory tab.
 - c. Click the Services node in the left pane.
 - d. Locate and click the DE node under Services.
 - e. Select Edit > Delete from the Directory Server menu bar (or right-click on the DE node and select Delete from the pop-up menu) to remove the DE node and all of its children.
7. To remove any links or references to the Directory Editor software, you must undeploy Directory Editor from your application server. Use the instructions provided in “Undeploying Directory Editor from Your Application Server” on page 168.

Uninstalling Manually From a Unix Machine

NOTE Use this procedure to uninstall Directory Editor *only* if the Directory Editor uninstaller is not operational.

Use the following steps to remove Directory Editor from a Unix machine:

1. Stop your application server.
2. Locate and open the `/bin` directory.
3. Double-click `shutdown.sh` to stop the Tomcat application.
4. Locate and open the `/webapps` directory.
5. Delete the `de.war` file and the `de` directory.
6. Remove the following `<compid>` elements and sub-elements from the `productregistry` file located in the `/var/opt/sun/install/` directory (on Linux) or from the `/var/sadm/install/` directory (on Solaris):

```
<compid>DE files
<compid>Directory Editor war file
<compid>Directory Editor
```

7. If necessary, return to the `/bin` directory and double-click `startup.sh` to restart the Tomcat application.
8. Remove configuration information from the Directory Server, which (by default) is located under `Services:DML`.
9. To remove any links or references to the Directory Editor software, you must undeploy Directory Editor from your application server. Use the instructions provided in “Undeploying Directory Editor from Your Application Server” on page 168.

Undeploying Directory Editor from Your Application Server

This section provides instructions for undeploying from the following application servers:

- “From Sun Application Servers” on page 168
- “From Tomcat Application Servers” on page 169
- “From WebLogic Application Servers” on page 169
- “From WebSphere Application Servers” on page 170

From Sun Application Servers

Use the following instructions to undeploy Directory Editor from a Sun Application Server:

1. Open the Sun ONE Administration Console.
2. In the left pane, open the `Web Apps` folder.
3. Locate the `de` node (or the name used when you deployed Directory Editor) and enable the checkbox next to that item.
4. Click the Undeploy button, and when you are prompted to confirm the action, click OK.

From Tomcat Application Servers

Use one of following methods to undeploy Directory Editor from a Tomcat Application Server:

- “Using the Tomcat Application Manager” on page 169
- “Undeploying Manually” on page 169

Using the Tomcat Application Manager

Use the following instructions to undeploy Directory Editor using the Tomcat Application Manger:

1. Open the Tomcat Application Manager.
2. In the Applications table, click the `de` link and then click the `Undeploy` link in the Command column.

Undeploying Manually

To undeploy Directory Editor manually from your Tomcat application server, use the instructions for uninstalling manually provided in “Removing the Software” on page 165.

From WebLogic Application Servers

Use the following instructions to undeploy Directory Editor from a WebLogic Application Server:

1. Open the WebSphere Console.
2. In the left pane, open the Deployments and Web Applications nodes.
3. Select `de` in the table and then click the trash can icon.
4. When you are prompted to confirm the action, click Yes.

From WebSphere Application Servers

Use the following instructions to undeploy Directory Editor from a WebSphere Application Server:

1. Open the WebSphere Administration Console.
2. Click the Enterprise Application link to display a list of applications.
3. Locate the de node and enable the checkbox next to that item.
4. Click the Stop button to stop the application.
5. Enable the de checkbox again, and then click the Uninstall button.

Error Logging and Troubleshooting

This chapter explains how to troubleshoot problems you may encounter as you work with Directory Editor. The information is organized into the following chapters:

- “Before You Call Technical Support” on page 171
- “Obtaining Information About the Problem” on page 172
- “Common Problems and Solutions” on page 177

Before You Call Technical Support

Before you call technical support, please get as much information as possible about the problem you are experiencing.

1. Using instructions provided in “Obtaining Information About the Problem” on page 172, to increase logging levels where appropriate and then try to re-create the problem.
2. Save all logs and error information, and be ready to provide this information to the technical support staff.

If you perform an action in Directory Editor and an error page displays, you can email that error page to your Technical Support representative. Directory Editor embeds exceptions as comments in its error pages.

3. Be prepared to answer questions about your Directory Editor installation. For example,
 - Which application server and version are you running?
 - Which J2SDK version are you running?
 - Which operating system is running the application server?

- Which browser version and operating system are you using to access Directory Editor?
- Which user was logged in when the problem occurred?
- How are the Startup Properties and Managed Directory configured?
- Which Directory Editor build are you running?

TIP To determine the build number, use one of the following methods:

- Hover your mouse cursor over the Version string at the top of the Directory Editor page. A tooltip message will display with the actual build number or (depending on your browser) the build number will be displayed as a message in the status bar (usually at the bottom of the browser window).
 - Right-click in the Directory Editor page and select View Page Source. When the source page is displayed, select Edit > Find in this page to search for "Build number."
-

Obtaining Information About the Problem

The first step to solving problems in Directory Editor is to figure out what went wrong. You can use information provided in this section to help diagnose and solve problems.

Information in the Page

If an error page displays in Directory Editor, the error page will contain information about the problem. (For example, Directory Editor embeds exceptions as comments in its error pages.) To locate this information, right-click on the page and when a pop-up menu displays, select View Page Source.

Inspect the page, and if it contains a Java stack trace or the exception information, select File > Save Page As from the browser's menubar to save the page, and then forward it to your Administrator for assistance.

Information in the Logs

There are various log files that may contain relevant information about the problem. These log files are described in the following sections:

- “Directory Editor Log” on page 173
- “Identity Manager Log” on page 174
- “Directory Server Log” on page 175
- “Application Server Logs” on page 176

Directory Editor Log

The Directory Editor log contains log statements generated from Directory Editor.

Possible log levels include:

- **FATAL:** Fatal error messages (*least detail*)
- **ERROR:** Non-fatal error messages
- **WARN:** Potentially harmful situations
- **INFO:** Informational messages highlighting application state
- **DEBUG:** Fine-grained informational events
- **TRACE:** Maximum detail

You can set these logging levels per Java class or package. For example, to get detailed messages about the operations being sent to the LDAP server, you can add the following line to the file:

```
log4j.logger.com.sun.dml.resource.LDAPConnector=TRACE
```

The `log4j.properties` file shipped with Directory Editor has more documentation about how to set logging levels, and some commonly used levels.

You can configure the log’s location and the logging levels in the `log4j.properties` file, which is located at

```
<application root>/WEB-INF/classes/log4j.properties
```

To change the log file location and logging levels, edit the `log4j.properties` file in a text editor. You must restart Directory Editor after saving your changes for the changes to take effect.

You can change the loggers in a log file to use a different appender. For example, the default `log4j.properties` file has an appender defined for a rolling log file that rolls with a maximum file size. To use this appender, you must change the `log4j.rootLogger` file to use the name of the new appender (for example, `log4j.rootLogger=WARN,rollingfile`).

For more information about configuring appenders and logging levels in the `log4j.properties` file, see the Log4j documentation provided at the following location:

<http://logging.apache.org/log4j/docs/>

Identity Manager Log

Some of Directory Editor's core components are provided by Sun Java™ System Identity Manager, which also supports logging. You can configure this product's logging using the Java system properties or the following file:

`<application root>/config/Waveset.properties`

Logging levels are configurable on a per package, per class, or per method basis. By default, the Identity Manager log goes to the `WSTrace.log` file located in the `<application root>/config/` directory. You can change this file by defining the `trace.destination` and `trace.Logging` levels as follows:

- **0:** Disable tracing
- **1:** Entry and exit of public methods, plus major exceptions
- **2:** Entry and exit of all methods
- **3:** Significant informational displays (such as variable values that control flow) that occur only once per method invocation
- **4:** Informational displays that occur *n* times per method invocation

Here as an example of tracing configuration in the `Waveset.properties` file:

Code Example 10-1 Sample Identity Manager Trace Configuration

```
trace.enabled=true
trace.level=0
trace.level.com.waveset.repository.ServerRepository=3
trace.level.com.waveset.repository.ServerRepository#init=0
trace.destination=STDOUT
# trace.destination=file
# trace.file=/var/log/WSTrace.log
```

You can also turn on tracing for forms and XPRESS by setting the `form.trace` and `xpress.trace` properties to `true` in the `Waveset.properties` file. You must restart the application server for any Identity Manager logging changes to take effect. For more information, see the *Sun Java™ System Identity Manager Technical Deployment Guide*.

Directory Server Log

Sun Java™ System Directory Server and Sun ONE Directory Server both can log directory access and detailed auditing messages.

The audit log contains detailed information about changes to the directory — such as creates, edits, deletes, and renames. The log entries provide information about:

- Time of the audit event
- DN of the affected object
- Type of change
- Specific attributes values that changed
- Who performed the change

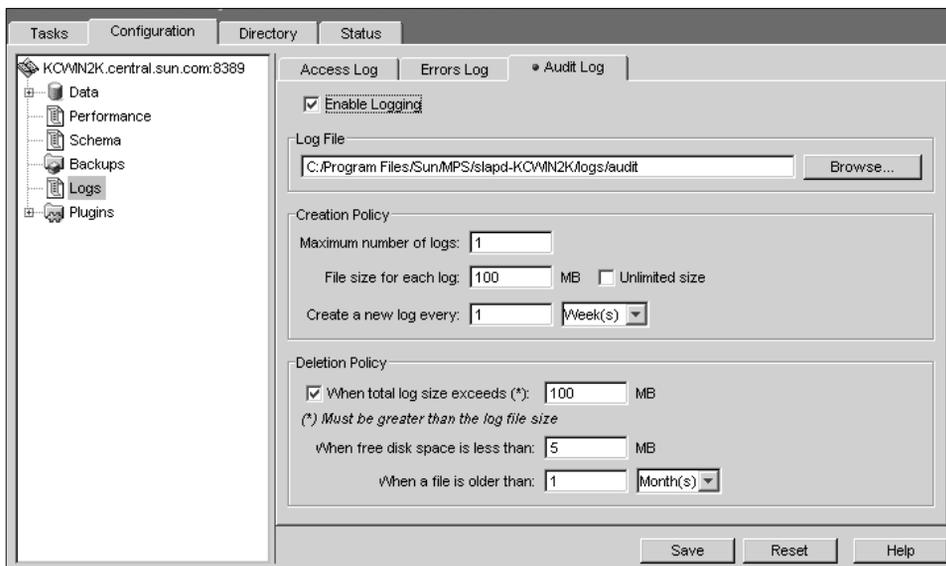
These log files are available at `<directory server root>/slapd-<host>/logs/` in the access and audit files respectively. (Directory Server keeps back-ups of these files in time-stamped files.)

The access file is enabled by default, and logs messages for directory binds, unbinds, searches, creates, edits, deletes, and renames.

The audit log file is not enabled by default. To enable audit logging, use the following steps:

1. Log into the Sun ONE Server Console.
2. Double-click a server name in the tree to open it.
3. Navigate to the Audit log configuration under Configuration > Logs > Audit Log.
4. Enable audit logging by enabling the Enable Logging check box.

Figure 10-1 Enabling Audit Logging



5. Click Save.

Application Server Logs

Generally, relevant information is available from one of the previously mentioned sources; but occasionally, error information will be logged in the application server's log files. The location and format of these messages varies, depending on which application server is being used to run Directory Editor.

Consult your application server's product documentation for information about accessing and configuring logs.

Common Problems and Solutions

This section describes some common problems you might encounter while using Directory Editor and explains how to remedy these problems.

- “`java.lang.OutOfMemoryError`” on page 177
- “Failed to Save Startup Configuration Properties” on page 178
- “`java.security.AccessControlException`” on page 179
- “WebSphere is Extremely Slow During Startup” on page 180
- “Changing the Object Class and Attributes Used for Configuration Objects” on page 181

Running in Non-Production Mode

You must configure the application server’s policy file to give Directory Editor permission to access the application server or Directory Editor will run in *non-production mode*, which means the Directory Editor cannot save its startup properties to the file system.

If you do not specify the appropriate write permissions, a message displays in the Directory Editor interface to let you know you are running in non-production mode.

Instructions for configuring a policy file for the different application servers are provided in “Step 3: Install Directory Editor on an Application Server” on page 13.

`java.lang.OutOfMemoryError`

An `OutOfMemoryError` message in the Directory Editor log file indicates the application server has run out of memory.

To increase the amount of memory available to the application server, you can add the `-Xmx<size>` setting to your JVM — using the `<size>` value to specify the maximum amount of memory the JVM is allowed to use. For example, `-Xmx512M` tells the server to use a maximum of 512 megabytes of memory.

Consult your application server’s product documentation for information about how to set command line arguments for the application server’s JVM.

Failed to Save Startup Configuration Properties

When Directory Editor is first started, you are prompted to enter some start-up properties that define where the configuration information for Directory Editor should be stored, how to bind to store the configuration information, and a few other options. Saving this information may fail if the application server cannot write to the `<application root>/WEB-INF/startup.properties` file.

You will encounter this problem if you have configured the application server to run this application in unexpanded mode or the application server's security features do not permit access to the host file system.

- **To start Directory Editor using the configuration you have specified:** Click Continue Without Saving to run the application in *non-production mode* (see “Running in Non-Production Mode” on page 177), which means your Directory Editor configuration will not be saved and you will be prompted to re-enter the configuration information every time you restart Directory Editor.
- **To fix this problem permanently:** If you are running from an unexpanded WAR (web application archive) file, extract and manually edit the `WEB-INF/startup.properties` file and then repackage the WAR file with the changed `startup.properties`.

NOTE You can also use this technique to edit the logging configuration mentioned in the previous section.

To manually edit the `startup.properties` file in an unexpanded WAR file:

- a. From a command prompt, navigate to the directory containing the `de.war` file.
- b. Unzip the `de.war` file with the following command (`jar` is installed with the Java Development Kit (J2SDK)):

```
jar xvf de.war
```

- c. Edit the `startup.properties` file to specify the required properties.
- d. Use the following command to update the `de.war` with the updated `startup.properties`:

```
jar uvf de.war WEB-INF/startup.properties
```

- e. Re-deploy the updated `de.war` file to the application server.

If you cannot save the updated `startup.properties` in an expanded WAR file, the application server does not have write access to the file. It might be necessary to give the user running the application server write access to `startup.properties`.

java.security.AccessControlException

Directory Editor requires access to some privileged operations that the application server may not allow by default. This problem will become apparent when a user first attempts to log into Directory Editor and sees the *Application Error* page.

The Directory Editor logs will show a `java.security.AccessControlException`.

To remedy this problem you must edit the security policy file for your application server and grant access to certain actions, as follows:

NOTE In Sun ONE Application Server you will find this file in the following location:

`<App Server Home>/domains/<Domain Name>/config/server.policy`

1. Edit the `server.policy` file.
2. Add the following lines to the end of the block granting access to “all remaining code”:

```
permission javax.security.auth.AuthPermission"getLoginConfiguration";
permission javax.security.auth.AuthPermission"setLoginConfiguration";
permission javax.security.auth.AuthPermission
"createLoginContext.SunDirectoryLogin";
permission javax.security.auth.AuthPermission"modifyPrincipals";
```

3. Save the file.
4. Restart the application server.

WebSphere is Extremely Slow During Startup

With some platforms, there is a performance impact if you use the JCE provided with that platform. If you experience a long start-up time, use Sun's JCE 1.2.2, which you can download from:

<http://java.sun.com/products/jce/>

Installation Instructions are provided at the following location:

http://java.sun.com/products/jce/jce122_install.html

If you install Sun JCE using the installation instructions provided, and you encounter exceptions during WAS startup, you may have to re-order the security providers specified in the following file so that the `IBMJCE` is first:

```
${java.home}/lib/security/java.security file
```

For example,

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
```

```
security.provider.2=sun.security.provider.Sun
```

```
security.provider.3=com.sun.crypto.provider.SunJCE
```

Modify the `${java.home}/lib/security/java.policy` file as follows:

```
grant codeBase "file:${java.home}/lib/ext/
sunjce_provider.jar" {
    permission java.io.FilePermission
        "${java.home}/jre/lib/ext/jce1_2_2.jar", "read";
    permission java.lang.RuntimePermission "getProtectionDomain";
    permission java.security.SecurityPermission "putProviderProperty.SunJCE";
};
```

Changing the Object Class and Attributes Used for Configuration Objects

By default, Directory Editor stores its configuration information in `applicationProcess` objects located in the configuration directory under:

```
ou=1.0,ou=DE,ou=Services,<base context>
```

If you want to change the location of these objects, the object class that is used to store the object, and/or the attributes used to store configuration information perform the following steps:

1. Back-up the existing configuration.
 - a. In Directory Editor, select Configure > Backup/Restore.
 - b. On the Backup/Restore page, click the Backup Configuration to File button.
 - c. When the Opening config.xml dialog box is displayed, enable the Save to Disk button to save the configuration XML file to your disk.
2. Delete the configuration organizational unit in the directory server.
 - a. Log into the Sun ONE Directory Server Console for the directory server that contains the configuration information.
 - b. Navigate to the configuration organizational unit in the directory at `ou=DML,ou=Services,<base context>`.
 - c. Right-click the DE entry and select Delete.
 - d. Click Yes to confirm deletion of the DE organizational unit.
3. Change the necessary properties in the `<application root>/WEB-INF/startup.properties` file.
 - a. To change the location where the configuration is stored, modify the `datastore.location` property so it points to the appropriate configuration organizational unit.
 - b. To change the object class used to store configuration object, modify the `datastore.objectClass` property to the appropriate object class.

NOTE You must define all required (*MUST*) attributes for the object class using the `datastore.dmlIdAttribute` or the `datastore.xmlObjectAttribute`.

- c. To change attributes in the object class used to store configuration information, modify the `datastore.dmlIdAttribute` and `datastore.xmlObjectAttribute`.
 - `deIdAttribute` contains a unique identifier used by Directory Editor
 - `xmlObjectAttribute` contains a (possibly large) XML text description of the object
4. Restart the application server.
5. Restore the configuration you backed up in Step 1.
 - a. In Directory Editor, select Configure > Backup/Restore.
 - b. On the Backup/Restore page, click the Browse button to locate and select the file you saved in Step 1. Click Open.
 - c. Click the Restore Configuration from File button to restore the file.

Short Session Timeout in Tomcat

If your Directory Editor session times out too soon, you can adjust the timeout period for Tomcat as follows:

1. Open the `web.xml` file located in `webapps/dml/WEB-INF/web.xml`.
2. Add the following lines to the `<web-app>` section, specifying a timeout value (in minutes) that is appropriate for your situation:

```
<session-config>
    <session-timeout>30</session-timeout>
</session-config>
```

de.war File Missing

If you are trying to upgrade or reinstall Directory Editor, and you did not uninstall the older product version first *or* you manually uninstalled Directory Editor, the Directory Editor installer will not provide the `de.war` file needed to complete the installation process.

NOTE You should not uninstall Directory Editor manually unless the Directory Editor uninstaller is not operational.

You must remove the following <compid> elements and sub-elements from the productregistry file:

```
<compid>DE files
<compid>Directory Editor war file
<compid>Directory Editor
```

The productregistry file is located in the following directory:

- **On Linux:** /var/opt/sun/install/
- **On Solaris:** /var/sadm/install
- **On Windows:** c:\winnt\system32\

HTML Components Used to Define Directory Editor Forms

This appendix describes the HTML components you can use when customizing Directory Editor forms.

Overview

To customize a Directory Editor form, you can use Directory Editor's Form XML language (also called forms), to describe HTML display components.

Directory Editor interprets this language at runtime to build the necessary components, allowing you to generate new pages dynamically with little or no additional Java development — which greatly simplifies customization.

This section contains the following topics:

- “What Are HTML Components?” on page 186
- “Component Classes” on page 186
- “Page Processor Requirements for HTML Components” on page 187
- “Component Subclasses” on page 188

What Are HTML Components?

HTML display components are instances of Java classes that generate a string of HTML text. Each display component has:

- **Class name** (defined in the field by the `DisplayClass` element): Identifies the component class, which determines the component's fundamental behavior and defines the set of properties recognized by the component.
- **Properties** (defined in the field with `Property` elements): Further define field behavior and appearance. You can specify one or more properties.

Specify display components as follows:

```
<Field name=' Name '>  
  <Display class=' Class '>  
    <Property name=' Name ' value=' Value '/>  
  </Display>  
</Field>
```

Component Classes

HTML components are independent objects that can be combined in various ways. Related components are organized into classes. There are two major groups of component classes:

- **Basic Component classes:** Components used to display and edit a single value.
- **Container classes:** Components that can contain one or more components.

Basic Component Classes

Common component classes include the components that are used to display and edit a single value. These components are defined in the section titled Basic Components.

Container Classes

A *container* class defines a collection of components that are visually organized in a certain way. Typically, creating a container class results in the generation of an HTML `table` tag. Simple containers can concatenate the components horizontally or vertically. Other containers allow more flexible positioning of components and may add ornamentation around the components.

Because containers are themselves components, any container can be placed inside another container. You can use this mechanism to build complex page layouts.

For example, many pages consist of a title, followed by a list of editing fields, followed by a row of form submission buttons. You can create a page by creating a `Panel` component using vertical orientation that contains a `Label` component, an `EditForm` component, and a `ButtonRow` component.

The `EditForm` component itself contains some number of subcomponents. The `ButtonRow` component is simply a `Panel` that uses horizontal orientation and contains a list of `Button` components.

Page Processor Requirements for HTML Components

This section describes the Directory Editor page processor requirements to display forms that implement HTML components. It covers:

Hidden Parameters

Most components have a name that corresponds to the name of a parameter posted from an HTML form. Directory Editor reserves a few parameter names for general use. Do not use these names as component names.

Table A-1 Hidden Parameter Names

Reserved Name	Description
<code>id</code>	Contains the ID of the object being edited
<code>command</code>	Contains the value of the button used to submit the form
<code>activeControl</code>	Contains the name of the last component that was active on the form

Table A-1 Hidden Parameter Names *(Continued)*

Reserved Name	Description
message	Can contain an informational message to be displayed at the top of the page
error	Can contain an error message to be displayed at the top of the page

Component Subclasses

All components extend the `Component` class, which defines the properties common to most components. In addition, some components extend the `Container` class, which gives them the ability to contain other components.

Each `Component` subclass defines a number of properties that are used to specify the characteristics of the component beyond those implied by the `Component` base class. For example, the `Label` component supports a `font` property, which can be used to specify the font used when rendering the label.

Naming Conventions

Properties always begin with a lowercase letter and use camel case to separate adjacent words. Access method names are formed by capitalizing the property name, and prefixing either `get` or `set`. For example, the property named `font` is accessible from Java using the following methods:

- `getFont`
- `setFont`

The data type for each property varies and is documented with the property. the terminology used to describe property value types is described in the following table.

Data Types

This table lists the data types allowed in component properties.

Table A-2 Component Properties Data Types

Reserved Name	Description
null	Indicates that a property has no value
string	<p>Represents the most common data type. String values are usually represented by an instance of the Java <code>String</code> class. Some components are values of any class. These are implicitly coerced to strings with the <code>toString</code> method.</p> <p>Unless otherwise specified, you can assume that all properties are of type <code>string</code>.</p> <p>Example:</p> <pre><String>Hello World</String></pre>
list of string	<p>Indicates that the value is expected to be a list of one or more strings. In Java, this value is always implemented as an instance of the <code>List</code> class. The elements of the list are then expected to be instances of the <code>String</code> class.</p> <p>Example:</p> <pre><List> <String>choice one</String> <String>choice two</String> </List></pre>
comma list	<p>Represents a single string that contains character ranges separated by a comma character. Components that have multi-valued properties typically allow either <code>list of string</code> values or <code>comma list</code> values. When a comma list is assigned to a property, the component converts it to a list of strings by extracting the substrings between commas and adding them to a list in the order in which they appear in the string.</p>

Table A-2 Component Properties Data Types *(Continued)*

Reserved Name	Description
boolean	<p>Represents a logically <code>true</code> or <code>false</code> value. Many properties require Boolean values.</p> <p>A Boolean value can be represented several ways. The <code>false</code> value represents one of the following values:</p> <ul style="list-style-type: none"> • A null value • A Boolean object whose value is <code>false</code> • An Integer object whose value is 0 • A string whose value is anything other than <code>true</code>. <p>When defining components in XML, Boolean values are most often represented with the strings <code>true</code> and <code>false</code>. (Any string value other than <code>true</code> is considered <code>false</code>.)</p>
int	<p>Represents a positive or negative integer. The <code>int</code> properties are commonly specified using strings. In that case, the string is coerced to an integer using the usual rules. If a string contains invalid integer characters, the coerced value is zero.</p>

Base Component Class

The `Component` class is the base class for all HTML components. It contains the properties that are common to most components. Not all `Component` properties are relevant in every subclass. For example, `Component` defines a property `allowedValues` that can contain a list of value constraints. This property is relevant only in subclasses that allow value editing such as `Select` or `MultiSelect`.

Further, `Container` classes almost never directly represent an editable value. Consequently, any properties related to the component value are irrelevant. Some properties are relevant only if the component is contained within a specific `Container` class.

This section describes the Base Component Classes used by Directory Editor:

- “name” on page 191
- “title” on page 191
- “value” on page 192
- “allowedValues” on page 192
- “primaryKey” on page 193

- “required” on page 193
- “noNewRow” on page 193
- “location” on page 194
- “help” on page 194
- “command” on page 194
- “onClick” on page 195
- “nowrap, align, width, and colspan” on page 196

name

Specifies the internal name of a field. All editing components must have a name, which is typically unique among all components displayed on the page. The `name` value is a string that is usually a path to a view attribute.

Container components do not require names and any assigned names are ignored. When building components from Java, component names are defined by the application. When building components from XML forms, component names are derived from the names of `Field` elements in the form. Field names are in turn path expressions within the view object that is used with the form.

Example:

```
<Field name = 'global.firstname'>
```

title

(Optional) Specifies the external name of a field. Titles are typically used with the `EditForm` container, which builds an HTML table that contains titles in one column and components in another.

Components do not render their own titles. Rendering of titles is controlled by the container. Many containers ignore titles.

Examples:

```
<Property name='title' value='FirstName' />
  <Property name='title'>
    <expression>
      <concat>
        <s>Edit User: </s>
        <ref>waveset.accountId</ref>
```

```

        </concat>
    </expression>
</Property>

```

In this example, the field title is in part derived dynamically from the user's Directory Editor account ID.

value

Editing components have a value that may be null. The value is typically set automatically by Directory Editor from an attribute in a view. Some components allow you to set the value by explicitly ignoring current view content. This value can be null.

The `Component` class allows the value to be any Java object. The subclass must coerce the value to a particular type when it is assigned, or when the HTML is generated. Component values are almost always `String` objects or `List` objects that contain strings. See "Data Types" on page 189 for more information about component value types.

Most container classes do not have values. If you assign a value, it is ignored. Some containers do allow values (for example, `TabPanel` and `WizardPanel`).

When building components from XML forms, the value is usually derived by using the component name as a path into the underlying view object, which contains all the values being edited.

Example:

```
<Property name= 'value' value='false' />
```

allowedValues

Specifies an optional list of allowed values for the component. If specified, the component allows you to select from only values that are on the list. If the component supports value restrictions, the list of allowed values is stored here. The value is always a list and usually contains strings. For convenience when setting properties from XML forms, you can also specify the allowed values as a comma list.

Example:

```
<Property name='allowedValues' value= 'Mon, Tue, Wed, Thurs, Fri' />
<Property name='allowedValues' />
    <expression>

```

```

    <call name='DaysoftheWeek' />
  </expression>
</Property>

```

primaryKey

This property is recognized only by the `SortingTable` container. The `SortingTable` container organizes components into a table with each column expected to contain components of the same class. `SortingTable` allows the rows to be sorted according to the values in any column. Typically, the sort order is determined from the value of each component in the column. There may be cases, however, where the value of the component is not suitable for sorting or may be inefficient to compare. In these cases, you can specify an alternate numeric sorting key.

required

If `true`, indicates that the field is expected to have a value before the form is submitted. If the component is contained within an `EditForm`, a red * (asterisk) will be placed after the component to indicate that the user must enter a value before saving. If the required schema map attribute is selected, (that is, set to a value of `true`), the field is always required.

The value of the property must be either `true` or `false`.

Example:

```
<Property name='required ' value='true '/>
```

noNewRow

If `true`, the field displays on the Directory Editor page next to the previous field. If not specified or set to `false`, the field appears on a new line, directly under the previous field. The default value is `false`.

This Boolean property is recognized only if the field is contained in a form that uses the `EditForm` display class. Typically, `EditForm` renders each component on a new row with the titles aligned in the left column and the component in the right column. To conserve space, you can concatenate several components on the same row. If the component also has a title, the title is rendered as non-highlighted text between the previous component and this component.

Values include:

```
value='true ' | 'false '
```

Example:

```
<Property name='noNewRow ' value='true '/>
```

location

Use if the container defines more than one display area and the component must be added to a specific area. Some containers allow the placement of components to be controlled by assigning a value to the `location` property.

For example, the `BorderedPanel` container supports five different display areas: north, south, east, west, and center.

The recognized values for the `location` property are defined by the container. If you do not assign a location, or assign a location name that is not recognized, the container places the component in the default location.

help

Specifies text that may be displayed to assist the user in understanding purpose of the field. In most Directory Editor pages, this will cause the `<icon>` icon to be displayed next to the component title. Moving the mouse over this icon will cause the help text to be displayed in the left margin.

The value of the property can either be literal text to be displayed, or it can be a message catalog key. Literal text can include HTML markup.

command

Specifies a command to submit when a component is modified. (When a user makes a change to a value, form output is recalculated.)

This property is typically used with the `Button` component. Some components must cause immediate submission of the surrounding HTML form when they are modified so that the application can regenerate the page based on that modification. Setting the `command` property to a non-null value causes this behavior.

When the `command` property is set, and the component is modified, the form is posted and an extra hidden parameter named `command` is posted whose value is the value of the `command` property.

The `command` specifies how the system will process the edits that have been made to a view. The `command` property must have one of the following values.

Table A-3 Command Property Values

Value	Description
Save	Causes the edits to be saved.
Cancel	Causes the edits to be discarded.
Recalculate	Causes the page to be regenerated.
SaveNoValidate	Causes the edits to be saved, but no form validation to be performed.

Because specifying a command value of `Recalculate` is so common in forms, a shorter alternative syntax is available. The `Display` element has an attribute named `action` that when set to `true`, has the same effect as setting the `command` property to *Recalculate*.

```
<Display class='Select' action='true'>
```

onClick

When specified, the value is expected to contain JavaScript that will be assigned as the value of the `onClick` attribute of the `input` element generated for this component. Not all components support the `onClick` property.

Use of this property is rare and requires detailed knowledge of the generated HTML. If you use this property, the page must typically contain a JavaScript component that defines JavaScript functions you call from within the `onClick` value.

Example:

```
<Property name='onClick' value="Uncheck(this.form,
'resourceAccounts.selectAll');"/>
```

NOTE Once forms are stored in the repository, Directory Editor always uses single quotes to surround attribute values. If single quotes appear within the attribute value, they will be replaced with `'`. To prevent this escaping you can represent the string in an XPRESS s expression:

```
<Property name='onClick'>
  <s>Uncheck(this.form, 'resourceAccounts.selectAll');</s>
</Property>
```

onChange

Similar to `command`. The value can be an arbitrary JavaScript statement to run when the field is modified.

Not all components support the `onChange` property.

Use of this property is rare and requires detailed knowledge of the generated HTML. If you use this property, the page must typically contain a `Javascript` component that defines JavaScript functions you call from within the `onChange` value.

nowrap, align, width, and colspan

Most containers position subcomponents by surrounding them with an HTML `table` tag. The HTML generated for each component then is typically contained in a `td` tag. Some containers can recognize the `nowrap`, `align`, `width`, and `colspan` properties and use them when generating the surrounding table cell tag. You can use these components to adjust the position and size of the component within the container.

The `nowrap` property affects how some components are displayed if they contain a long string of text. If the value of `nowrap` is `false` or unspecified, the browser can break up the component text into multiple lines when it is displayed. If the value of `nowrap` is `true`, the browser will try to keep the component text on a single line.

The `align` property is rarely used. You can use `align` to adjust the element horizontally on the form. Allowed values are `left`, `right`, and `center`.

Use of the `colspan` property is deprecated.

Examples

```
<Property name='width' value='3' />  
<Field name='Start Day' prompt='Day' nowrap='true' />
```

Basic Components

This section describes the Basic Components used by Directory Editor:

- “BackLink” on page 197
- “Button” on page 198
- “DatePicker” on page 199
- “Javascript” on page 199
- “Label” on page 200
- “Link” on page 201
- “MultiSelect” on page 201
- “Radio” on page 203
- “SectionHead” on page 204
- “Select” on page 204
- “SimpleTable” on page 205
- “Text” on page 205
- “Text Area” on page 206
- “SimpleTable” on page 205

BackLink

Displays a link that returns to the previous page. The behavior of this component is the same as that of the browser Back button. However, you may want to place this link in a convenient position on the page.

The only property for this display component is `text`, which is used to specify text for the link. If you do not specify `text`, the link defaults to `Back`.

Example:

```
<Field name='back'>
  <Display class='BackLink'>
    <Property name='value' value='previous page' />
  </Display>
</Field>
```

Button

Displays a button.

Properties for this display component are:

- **name:** Specifies the name of the parameter that will be posted when the user clicks this button. This property is optional; if not specified, the default value is `command`.
- **value:** Specifies the value of the parameter posted when the user clicks this button.
- **label:** Specifies the visible text that displays on the button.
- **command:** Specifies an optional value to submit along with the `name` parameter (for example, Save, Cancel, Recalculate).
- **postURL:** Specifies an alternate, target URL to which the form will be posted. This value overrides the URL specified in the JSP.
- **hiddenID:** Specifies an optional value for an id parameter to be included in the form post data.

Example:

```
<Display class='Button'>
  <Property name='label' value='Change Password' />
  <Property name='value' value='Recalculate' />
</Display>
```

Checkbox

Displays a checkbox.

- When enabled, the box represents a value of `true`.
- When disabled, the box represents a `false` value.

Properties for this display component are:

- **label:** (Optional) Specifies a label that is displayed to the right of the checkbox. It is displayed adjacent to the component, but is not displayed in the title column
- **leftLabel:** Specifies that the label should appear to the left of the checkbox.
- **value:** Determines the state of the checkbox. If the value is logically true, the checkmark is displayed.

Example:

```
<Field name='accounts[NT].passwordExpired'>
  <Display class='checkbox'>
    <Property name='title' value='Password is Expired' />
  </Display>
</Field>
```

DatePicker

Allows the user to specify a date using an applet that displays a calendar. The field is displayed in the form as a calendar icon. When the icon is clicked, the calendar applet is launched in a separate window.

Properties for this display component are:

The only property for this display component is `value`, which specifies the date to be highlighted on the calendar as the current date. You can parse the `Date` property from either a `Date` object or a `String` object. `String` values are expected to conform to the "MM/dd/yyyy" format.

Example:

```
<Field name='ExpireDate'>
  <Display class='DatePicker'>
    <Property name='title' value='Set Password Expire date' />
  </Display>
</Field>
```

Javascript

Use to insert pre-formatted JavaScript into the page. This is useful if you are using the `onClick` or `onChange` properties in components and want to call custom JavaScript functions.

Though not required, you should specify the `name` property when building components from XML forms. Using features such as field loops and field inclusion, you can add more than one JavaScript component containing the same script to the page. During HTML generation, JavaScript components that have the same name are included only once.

The component has an extended property named `script` that can contain the JavaScript text.

You can also include JavaScript by setting the property named `source`. This should be a string containing a URL fragment relative to the base context. It is the JavaScript contained in the indicated file to be loaded by the browser.

Label

Displays a string of text.

Properties for this display component are:

- **value:** Defines the text to be displayed. The value can be either a string or a list of strings. When the value is a list, each string in the list is displayed on a separate line.
- **leftPad:** Specifies the number of spaces to insert to the left of the label.
- **pad:** Specifies the number of spaces to insert to the left and right of the label.
- **rightPad:** Specifies the number of spaces to insert to the right of the label.

NOTE If no padding is specified, the default padding is `leftPad=2,`
`rightPad=2.`

Example:

```
Title<Field>
  <Display class='Label'>
    <Property name='title' value='Account ID' />
    <Property name='value'>
      <ref>waveset.accountId, /ref>
    </Property>
  </Display>
</Field>
```

- **font:** Specifies the font style. The value must be one of the styles defined in the `styles/style.css` file of the Directory Editor installation directory.
- **color:** Specifies the label color. Use standard HTML color formatting (`# xxxxxx`) to specify the color value.

Link

Places a link on the page.

Properties for this display component are:

- **URL:** Specifies the target Uniform Resource Locator (URL).
- **imageURL:** (*optional*) Specifies the URL to an icon or image that will be rendered to the right of the link.
- **imageURL2:** (*optional*) Specifies the URL to an icon or image used will be rendered to the right of the first image.
- **hoverText:** Specifies text to display when the mouse rests over the first or second image.
- **id:** (*optional*) Specifies a value to be included as the id query argument in the link.
- **arguments:** (*optional*) Specifies a set of name/value pairs to be included as query arguments.
- **extraURL:** (*optional*) Specifies an additional URL fragment to be included after the base URL and arguments.

Example:

```
<Field>
  <Display class='Link'>
    <Property name='name' value='Request Group Access' />
    <Property name='URL' value='user/processLaunch.jsp?newView=true'>
    <Property name='id' value='Group Request Process' />
  </Display>
</Field>
```

MultiSelect

Displays a multiple-selection text box, which displays as a two-part object in which a defined set of values in one box can be moved to a selected box. Values in the left box are defined by the `allowedValues` property, values are often obtained dynamically by calling a Java method such as `FormUtil.getResources`. The values displayed in the right side of a multiple-selection box are populated from the current value of the associated view attribute, which is identified through the field name.

The form titles for this two-part object are set through the `availabletitle` and `selectedtitle` properties.

Properties for this display component are:

- **availableTitle:** Specifies the title of the available box.
- **selectedTitle:** Specifies the title of the selected box.
- **ordered:** Defines whether selected items can be moved up or down within the list of items in the text box. A true value indicates that additional buttons will be rendered to permit selected items to be moved up or down.
- **allowedValues:** Specifies the values associated with the left side of the multiple-selection box. This value must be a list of strings.

NOTE The *<Constraints>* element can be used to populate this box, but its use is deprecated.

- **sorted:** Specifies that the values in both boxes will be sorted alphabetically.
- **noApplet:** Specifies whether the multiselect will be implemented with an applet or with a pair of standard HTML select boxes. The default is to use an applet, which is better able to handle long lists of values.
- **typeSelectThreshold:** (Available only when the `noApplet` property is set to true.) Controls whether a type-ahead select box appears under the `allowedValue` list. When the number of entries in the left select box reaches the threshold defined by this property, an additional text entry field appears under the select box. As you type characters into this text field, the select box will scroll to display the matching entry if one exists. For example, if you enter `w`, the select box scrolls to the first entry that begins with `w`.
- **lockhart:** Controls the look-and-feel of the non-applet version of this component.
 - Specify `true` so the look-and-feel of this component is consistent with other Directory Editor components.
 - If you do not specify a value, the browser default will be used.

Example:

```
<Field name='accounts[LDAP].LDAPDept' type='string'>
  <Display class='MultiSelect' action='true'>
    <Property name='title' value='LDAP Department' />
  </Display>
  <Constraints>
    <0>
```

```

        <List>
            <String>Sales</String>
            <String>Marketing</String>
            <String>International Sales</String>
        </List>
    </o>
</Constraints>
</Field>

```

Radio

Displays a horizontal list of one or more radio buttons. A user can select only one radio button at a time. If the component value is null or does not match any of the allowed values, no button is selected.

Properties for this display component are:

- **title:** Specifies the title for all radio buttons.
- **labels:** Specifies an alternate list of button labels. The labels list must be as long as the values in the `allowedValues` list. Alternate labels can be used in cases where the values are cryptic. For example, values can be letter codes such as "H", "M", and "S", but you would use this property to identify button labels "hours", "minutes", and "seconds".
- **allowedValues:** Specifies the value associated with each button. This value must be a list of strings.
- **value:** Specifies values for the buttons. This value accepts one string. If not set, then the values are the same as the labels.

Example:

To display a container that contains two fields that prompt the user for a delayed start time for an activity, specify the following code. It will create a text field in which the user enters a numeric value and a second field of radio buttons that the user sets to specify the increment. The field has three radio buttons labeled (in order) Days, Hours, and Minutes, with values of minutes, hours, days, weeks, and months.

The `StartDate` field converts the value specified by the user into hours.

The only property for this display component is `value`, which specifies the date to be highlighted on the calendar as the current date. Date can be parsed from either a Date object or a String object. String values are expected to conform to the "MM/dd/yyyy" format.

Example:

```
<Field name='ExpireDate'>
  <Display class='DatePicker'>
    <Property name='title' value='Set Password Expire date' />
  </Display>
</Field>
```

SectionHead

Displays a new section heading defined by the value of the `text` property. It is an extension of the `Label` class that sets the font property to a style that results in large bold text. It also sets the `pad` property to zero to eliminate the default two-space padding. Use it to break up long forms into sections separated by a prominent label.

The only property for this display component is `text`, which specifies the text to be displayed.

Example:

```
<Field>
  <Display class='SectionHead'>
    <Property name='text' value='Calculated Fields' />
  </Display>
</Field>
```

Select

Displays a single-selection list box. Values for the list box must be supplied by the `allowedValues` property.

Properties for this display component are:

- **allowedValues:** Specifies the list of selectable values for display in the list box.
- **autoSelect:** When set to `true`, this property causes the first value in the `allowedValues` list to be automatically selected if the initial value for the field is null.
- **multiple:** When set to `true`, allows more than one value to be selected.
- **nullLabel:** Specifies the text that displays at the top of the list box when no value is selected.
- **size:** (*optional*) Specifies the maximum number of rows to display. If the number of rows exceeds this size, a scroll bar is added.

- **sorted:** When set to `true`, causes the values in the list to be sorted.
- **valueMap:** Maps raw values to displayed values.

The component supports the `command` and `onChange` properties.

Example:

```
<Field name='city' type='string'>
  <Display class='Select'>
    <Property name='title' value='City' />
    <Property name='allowedValues'>
      <List>
        <String>Austin</String>
        <String>Portland</String>
        <String>New York</String>
      </List>
    </Property>
  </Display>
</Field>
```

SimpleTable

Arranges components in a grid with an optional row of column titles at the top. The only property for this display component is `columns`, which assigns column titles and defines the width of the table as defined in a list of strings.

Text

Displays a regular text entry box.

Common properties for this display component are:

- **size:** Specifies the number of characters that are visible in the text entry box. The box size is recalculated depending upon the length of the text in the box.
- **maxLength:** Specifies the maximum length of the string that can be edited in the text box.
- **secret:** Displays `*****` (asterisks) in the place of entered text. This option is most often used in password fields.
- **readOnly:** Displays read-only text. This text cannot be edited by the user. You might use this property if, for example, you want to display resource attribute information that an administrator needs to view when creating or editing user accounts.

- **submitOnEnter:** When you set this property and the Text field has focus, then when the user presses the Enter key, the form is submitted using the command that is specified in the property value. In the following example, the form is submitted exactly as though the user has clicked the Save button.
- **multivalued:** When you set this property, the component is rendered with controls that allow users to add or remove multiple values of an attribute.

Example:

```
<Field name='variables.identityID'>
  <Display class='Text'>
    <Property name='required'>
      <Boolean>true</Boolean>
    </Property>
    <Property name='title' value='Identity ID' />
    <Property name='size' value='32' />
    <Property name='maxLength' value='128' />
    <Property name='submitOnEnter' value='Save' />
  </Display>
</Field>
```

Text Area

Displays a multi-line text entry box.

Properties for this display component are:

- **rows:** Specifies the number of text area rows.
- **columns:** Specifies the number of text area columns.
- **readOnly:** Displays read-only text in the text entry box.

Example:

To display a text box with five visible rows that wraps after each 70 characters specify:

```
<Field name='Description'>
  <Display class='TextArea'>
    <Property name='rows' value='5' />
    <Property name='columns' value='70' />
  </Display>
</Field>
```

If the user enters text beyond the defined visible rows, the text area displays a scroll bar.

Resources for Capability Configuration

Capabilities aggregate sets of actions and views that are necessary to complete a task in Directory Editor.

Capabilities are associated with resources using the mapping present in the “Default Capabilities” Configuration XML object that is stored in the configuration directory. The configuration enumerates all capabilities, each of which contains a list of associated resources.

Currently, all of these resources are either *actions* or *views*.

- **Actions** are represented by a string such as `/Edit`.
- **Views** are represented by a string is similar to `.nav_tab.Edit`.

NOTE Typically, each tab view is represented using both an *unselected* representation and a *selected* representation, such as `.nav_tab_selected.Edit`.

Table B-1 Resources Available for Role Configuration

Resource Name	Type	Category	Description
/AddGroupMembersSearch	Struts Action	Object Edit/ Create	Displays a form for adding members to a group object
/BootstrapEditor	Struts Action	Initialization	Edit Startup Properties
/CloseAndCancelWindow	Struts Action	Core	Handles pop-up dialog cancel actions
/CloseAndSaveWindow	Struts Action	Core	Handles pop-up dialog save actions
/ConfigImpExp	Struts Action	Configuration	Displays a form for backing up and restoring the configuration to and from file
/ConfigureManagedDirectory	Struts Action	Configuration	Displays a form for editing of managed directory properties during runtime
/Create	Struts Action	Object Edit/ Create	Displays a form for editing object attributes before creation
/CreateFromList	Struts Action	Object Edit/ Create	Displays a form for editing object attributes before creation; returning to the directory tree when finished
/CreateOptions	Struts Action	Object Edit/ Create	Displays a form for specifying the type and location for a new object
/CreateOptionsFromList	Struts Action	Object Edit/ Create	Displays a form for specifying the type and location of a new object; returning to the directory tree when finished
/CreateRole			Displays a form that allows the user to create an application role
/Debug	Struts Action	Debug	Enables the Debug tab
/Debug/CalledByMethod			Shows CallTimer for a specific method
/Debug/CallTimer			Shows CallTimer statistics
/Debug/CallTimerExport			Exports CallTimer information to XML
/Debug/CallTimerImport			Imports CallTimer information to XML
/Debug/Config	Struts Action	Debug	Forward user to a page showing startup properties

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
/Debug/HTTPSession	Struts Action	Debug	Forward user to a page showing current HTTP session state
/Debug/Memory	Struts Action	Debug	Forward user to a page showing current memory usage
/Delete	Struts Action	Object Edit/ Create	Display a form prompting the user form confirmation before deleting an object
/DeleteFromList	Struts Action	Object Edit/ Create	Display a form prompting the user form confirmation before deleting an object; returning to the directory tree when finished
/Dialog	Struts Action	Core	Handle pop-up dialog interactions
/Disable	Struts Action	Object Edit/ Create	Display a form prompting the user form confirmation before disabling an object
/DisableFromList	Struts Action	Object Edit/ Create	Display a form prompting the user form confirmation before disabling an object; returning to the directory tree when finished
/Edit	Struts Action	Object Edit/ Create	Display a form for editing an existing object's attributes
/EditPassword	Struts Action	Core	Display a form for editing the user's own password
/EditRole			Displays a form that allows the user to edit an application role
/Enable	Struts Action	Object Edit/Create	Display a form prompting the user form confirmation before enabling an object
/EnableFromList	Struts Action	Object Edit/ Create	Display a form prompting the user form confirmation before enabling an object; returning to the directory tree when finished
/EndBootstrapMode	Struts Action	Initialization	Sets internal system state indicating that the application is ready for use
/Error	Struts Action	Core	Forward user to a generic error dialog
/ExportConfiguration	Struts Action	Configuration	Writes the current configuration to an output stream

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
/ExportGroupMembers	Struts Action	Object Edit/ Create	Download a newline-delimited file of the member DNs of a given group
/ExportLDIF	Struts Action	Configuration	Download an LDAP file of a given object
/FindPrincipals			Displays a form that allows the user to search for LDAP objects to add as role principals
/Forbidden	Struts Action	Core	Forward user to an error dialog telling them they are forbidden from accessing the requested resource
/Help	Struts Action	Core	Forward user to the online help browser
/Home	Struts Action	Primary Navigation	Forward user to the home page
/Initialize	Struts Action	Initialization	Initializes the application upon startup verifying startup configuration
/InitializeConfig	Struts Action	Initialization	Forward user to a page for specifying startup properties
/InitializeConfigDirSubmit	Struts Action	Initialization	Handle startup properties
/InitializeConfigSaveError	Struts Action	Initialization	Forward user to a dialog saying that there was a problem persisting startup properties
/InitializeManagedDirectory	Struts Action	Initialization	Display a form for editing managed directory properties during initialization
/InlineDialogResponse	Struts Action	Core	Handles response from inline dialogs
/List	Struts Action	Primary Navigation	Forward user to the directory tree
/Login	Struts Action	Core	Forward user to log in page
/LoginSubmit	Struts Action	Core	Handles log in parameters
/LogoutSubmit	Struts Action	Core	Handles log out parameters
/ManageForms	Struts Action	Configuration	Display a form allowing the user manage system forms

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
/ManageRoles	Struts Action	Configuration	Display a form allowing the user manage application roles
/NamingAttributeConfig	Struts Action	Configuration	Display a form allowing the user specify object naming policy
/NoManagersWarning	Struts Action	Initialization	Forward user to a dialog warning the user that the manager group has no members
/PasswordExpired	Struts Action	Core	Forward user to a dialog saying that the user's password has expired
/PasswordExpiring	Struts Action	Core	Forward user to a dialog saying that the user's password will expire and giving them the option to change it
/RemoveGroupMembersSearch	Struts Action	Object Edit/ Create	Display a form allowing the user to remove members from a group
/Rename	Struts Action	Object Edit/ Create	Display a form prompting the user form confirmation before renaming an object
/RenameFromList	Struts Action	Object Edit/ Create	Display a form prompting the user form confirmation before renaming an object; returning to the directory tree when finished
/Search	Struts Action	Primary Navigation	Forward user to a page allowing them to search the directory
/SearchConfig	Struts Action	Configuration	Display a form allowing the user specify properties the application will use for searching the directory
/SetApplicationMode	Struts Action	Initialization	Sets internal system state indicating the mode at which the application is operating
/ShowGroupMembersSearch	Struts Action	Object Edit/ Create	Display a form allowing the user to display the current members from a group
/Unavailable	Struts Action	Core	Forward user to a dialog saying that the application is currently unavailable
.nav_bar.Base	Struts Tile	Core	Base type for navigation bars
.nav_bar.Configure	Struts Tile	Configuration	Navigation bar with the configuration tab selected

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
.nav_bar.Create	Struts Tile	Object Edit/ Create	Navigation bar with the create tab selected
.nav_bar.Debug	Struts Tile	Debug	Navigation bar with the debug tab selected
.nav_bar.Home	Struts Tile	Primary Navigation	Navigation bar with the home tab selected
.nav_bar.List	Struts Tile	Primary Navigation	Navigation bar with the directory tree tab selected
.nav_bar.Search	Struts Tile	Primary Navigation	Navigation bar with the search tab selected
.nav_tab_selected.Base	Struts Tile	Core	Base type for selected navigation bar tabs
.nav_tab_selected.Configure	Struts Tile	Configuration	Selected navigation tab for configure
.nav_tab_selected.Create	Struts Tile	Object Edit/Create	Selected navigation tab for create
.nav_tab_selected.Debug	Struts Tile	Debug	Selected navigation tab for debug
.nav_tab_selected.Home	Struts Tile	Primary Navigation	Selected navigation tab for home
.nav_tab_selected.List	Struts Tile	Primary Navigation	Selected navigation tab for the directory tree
.nav_tab_selected.Search	Struts Tile	Primary Navigation	Selected navigation tab for search
.nav_tab.Base	Struts Tile	Core	Base type for unselected navigation bar tabs
.nav_tab.Configure	Struts Tile	Configuration	Unselected navigation tab for configure
.nav_tab.Create	Struts Tile	Object Edit/ Create	Unselected navigation tab for create
.nav_tab.Debug	Struts Tile	Debug	Unselected navigation tab for debug
.nav_tab.Home	Struts Tile	Primary Navigation	Unselected navigation tab for home
.nav_tab.List	Struts Tile	Primary Navigation	Unselected navigation tab for the directory tree

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
.nav_tab.Search	Struts Tile	Primary Navigation	Unselected navigation tab for search
.page.AddGroupMembersSearch	Struts Tile	Object Edit/Create	Page displaying a form for adding members to a group
.page.AuthBase	Struts Tile	Core	Base type for application pages requiring authentication for access
.page.Base	Struts Tile	Core	Base type for application pages
.page.Bootstrap	Struts Tile	Initialization	Page for specifying startup properties at initialization time
.page.BootstrapEditor	Struts Tile	Configuration	Page for specifying startup properties at run time
.page.BootstrapSaveError	Struts Tile	Initialization	Dialog saying that there was a problem persisting startup properties
.page.CloseAndCancelWindow	Struts Tile	Core	Page for handling pop-up dialog cancel requests
.page.CloseAndSaveWindow	Struts Tile	Core	Page for handling pop-up dialog save requests
.page.ConfigImpExp	Struts Tile	Configuration	Page displaying a form for backup and restore of configuration data
.page.Create	Struts Tile	Object Edit/Create	Page displaying a form for editing object attributes before creation
.page.CreateRole			Page displaying a form for creating a new role
.page.Debug.CalledByMethod			Page showing call timing information for a specific method
.page.Debug.CallTimer			Page showing call timing information
.page.Debug.CallTimerImport			Page for importing call timing information from XML
.page.Debug.Config	Struts Tile	Debug	Page showing startup properties
.page.Debug.HTTPSession	Struts Tile	Debug	Page showing current HTTP session state
.page.Debug.Memory	Struts Tile	Debug	Page showing current memory usage

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
.page.Delete	Struts Tile	Object Edit/Create	Page displaying a form prompting the user form confirmation before deleting an object
.page.Dialog	Struts Tile	Core	Page for displaying information in a pop-up
.page.Disable	Struts Tile	Object Edit/ Create	Page displaying a form prompting the user form confirmation before disabling an object
.page.Edit	Struts Tile	Object Edit/ Create	Page displaying a form for editing an existing object's attributes
.page.EditPassword	Struts Tile	Core	Page displaying a form for editing the user's own password
.page.EditRole			Page displaying a form for editing a role
.page.Enable	Struts Tile	Object Edit/ Create	Page displaying a form prompting the user form confirmation before enabling an object
.page.Error	Struts Tile	Core	A generic error dialog
.page.FindPrincipals			Page displaying a form for adding members to a group
.page.Forbidden	Struts Tile	Core	Error dialog telling the user they are forbidden from accessing the requested resource
.page.Forms	Struts Tile	Configuration	Page displaying a form for management of system forms
.page.Help	Struts Tile	Core	Page displaying the help viewer
.page.Home	Struts Tile	Primary Navigation	Page displaying a form for the user's home page
.page.InitializeManagedDirectory	Struts Tile	Initialization	Page displaying a form for specification of managed directory properties during initialization
.page.Initializing	Struts Tile	Core	Page displaying a wait message for use during initialization
.page.InlineDialog	Struts Tile	Core	Base page for supporting inline dialogs

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
.page.List	Struts Tile	Primary Navigation	Page displaying a form representing the directory tree
.page.Login	Struts Tile	Core	Page prompting for log in credentials
.page.ManagedDirectory	Struts Tile	Initialization/ Configuration	Page displaying a form for editing managed directory properties during run time
.page.NamingAttributeConfig	Struts Tile	Configuration	Page displaying a form allowing the user specify object naming policy
.page.NoManagersWarning	Struts Tile	Initialization	A dialog warning the user that the manager group has no members
.page.PasswordExpired	Struts Tile	Core	A dialog saying that the user's password has expired
.page.PasswordExpiring	Struts Tile	Core	A dialog saying that the user's password will expire and giving them the option to change it
.page.Popup	Struts Tile	Core	Base page for supporting pop-up dialogs
.page.RemoveGroupMembersSearch	Struts Tile	Object Edit/ Create	Page displaying a form for removing members from a group
.page.Rename	Struts Tile	Object Edit/ Create	Page displaying a form prompting the user form confirmation before renaming an object
.page.Roles	Struts Tile	Configuration	Page displaying a form for management of system roles
.page.Search	Struts Tile	Primary Navigation	Page displaying a form for searching the directory
.page.SearchConfig	Struts Tile	Configuration	Page displaying a form for configuration of application directory searches
.page.Secondary	Struts Tile	Core	Base page for content secondary pages
.page.ShowGroupMembersSearch	Struts Tile	Object Edit/ Create	Page displaying a form for showing members of a group
.page.Unavailable	Struts Tile	Core	Error dialog telling user that the application is currently unavailable
.subnav_bar.Base	Struts Tile	Core	Base type for sub-navigation bars

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
.subnav_bar.BootstrapEditor	Struts Tile	Configuration	Sub-navigation bar with the startup properties tab selected
.subnav_bar.ConfigImpExp	Struts Tile	Configuration	Sub-navigation bar with the configuration backup/restore tab selected
.subnav_bar.Debug.CallTimer			Sub-navigation bar with the debug tab showing call timing
.subnav_bar.Debug.Config	Struts Tile	Debug	Sub-navigation bar with the debug tab showing startup properties selected
.subnav_bar.Debug.HTTPSession	Struts Tile	Debug	Sub-navigation bar with the debug tab showing HTTP session state selected
.subnav_bar.Debug.Memory	Struts Tile	Debug	Sub-navigation bar with the debug tab showing current memory usage selected
.subnav_bar.Forms			Sub-navigation bar with the forms management tab selected
.subnav_bar.ManagedDirectory	Struts Tile	Configuration	Sub-navigation bar with the managed directory tab selected
.subnav_bar.NamingAttributeConfig	Struts Tile	Configuration	Sub-navigation bar with the naming attribute configuration tab selected
.subnav_bar.Roles	Struts Tile	Configuration	Sub-navigation bar with the role management tab selected
.subnav_bar.SearchConfig	Struts Tile	Configuration	Sub-navigation bar with the Search Options tab selected
.subnav_divider	Struts Tile	Core	The tab divider
.subnav_tab_selected.Base	Struts Tile	Core	Base type for selected sub-navigation tabs
.subnav_tab_selected.BootstrapEditor	Struts Tile	Configuration	Selected sub-navigation tab for startup properties
.subnav_tab_selected.ConfigImpExp	Struts Tile	Configuration	Selected sub-navigation tab for backup/restore of configuration
subnav_tab_selected.Debug.CallTimer			Sub-navigation bar with the debug tab showing call timing selected
.subnav_tab_selected.Debug.Config	Struts Tile	Debug	Selected sub-navigation tab for showing startup properties for debug

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
.subnav_tab_selected.Debug.HTTPSession	Struts Tile	Debug	Selected sub-navigation tab for showing HTTP session state for debug
.subnav_tab_selected.Debug.Memory	Struts Tile	Debug	Selected sub-navigation tab for showing current memory usage for debug
.subnav_tab_selected.Forms	Struts Tile	Configuration	Selected sub-navigation tab for managing system forms
.subnav_tab_selected.ManagedDirectory	Struts Tile	Configuration	Selected sub-navigation tab for specifying managed directory properties
.subnav_tab_selected.NamingAttributeConfig	Struts Tile	Configuration	Selected sub-navigation tab for managing naming attributes
.subnav_tab_selected.Roles	Struts Tile	Configuration	Selected sub-navigation tab for managing system roles
.subnav_tab_selected.SearchConfig	Struts Tile	Configuration	Selected sub-navigation tab for specifying search properties
.subnav_tab.Base	Struts Tile	Core	Base type for unselected sub-navigation tabs
.subnav_tab.BootstrapEditor	Struts Tile	Configuration	Unselected sub-navigation tab for startup properties
.subnav_tab.ConfigImpExp	Struts Tile	Configuration	Unselected sub-navigation tab for backup/restore of configuration
.subnav_tab.Debug.Config	Struts Tile	Debug	Unselected sub-navigation tab for showing startup properties for debug
.subnav_tab.Debug.HTTPSession	Struts Tile	Debug	Unselected sub-navigation tab for showing HTTP session state for debug
.subnav_tab.Debug.Memory	Struts Tile	Debug	Unselected sub-navigation tab for showing current memory usage for debug
.subnav_tab.Forms	Struts Tile	Configuration	Unselected sub-navigation tab for managing system forms
.subnav_tab.ManagedDirectory	Struts Tile	Configuration	Unselected sub-navigation tab for specifying managed directory properties
.subnav_tab.NamingAttributeConfig	Struts Tile	Configuration	Unselected sub-navigation tab for managing naming attributes

Table B-1 Resources Available for Role Configuration *(Continued)*

Resource Name	Type	Category	Description
.subnav_tab.Roles	Struts Tile	Configuration	Unselected sub-navigation tab for managing system roles
.subnav_tab.SearchConfig	Struts Tile	Configuration	Unselected sub-navigation tab for specifying search properties

Improving Performance of Browse and Search Features

This appendix explains how you can improve performance for Directory Editor's browse and search features.

The information is organized as follows:

- “Introduction” on page 220
- “Increasing Virtual List View Performance” on page 220
- “Creating a VLV Index” on page 220
- “Creating Tree Indexes” on page 226
- “Using the Virtual List View Anonymously” on page 229

NOTE The information in this appendix is provided to supplement the detailed information about browsing and searching indexes provided in the Sun Java™ System Directory Server product documentation.

If you encounter any information that conflicts with the Directory Server product documentation, you should consider the Directory Server documentation as authoritative.

Introduction

Directory Editor's browse and search features both use an LDAP protocol facility called the *Virtual List View*. This Virtual List View facility provides an LDAP client with the ability to specify a "window" into the data that is available on the LDAP server.

A client provides an LDAP query and sorting parameters that enable the server to create a "view" into the data that would be produced by the query. The client also provides the server with parameters that control which range of entries to retrieve. Because this facility places a significant load on the LDAP server, consider making the facility more performant if you will be using it regularly.

Increasing Virtual List View Performance

You can make the virtual list view (VLV) more efficient by creating a *VLV index* inside the directory server. A VLV index effectively notifies the server that a virtual list view, with specific query and sort parameters, will be performed. This index also allows the server to collect and maintain the information required to make using the virtual list view faster.

Be aware that there are several small costs associated with indexing:

- Because browsing indexes must be stored in persistent storage, more disk space will be required for the database.
- Because insert, delete, and update operations may affect the indexes, there may be a small decrease in write performance.

In most deployments however, the increase in query speeds attained by indexing far outweigh these costs.

Creating a VLV Index

This section provides instructions for creating a VLV index

- "From the Directory Server Console" on page 221
- "From the Command Line" on page 225

NOTE Based on the VLV's performance, consider using a VLV index for an `ou` with more than 500 to 1000 entries.

From the Directory Server Console

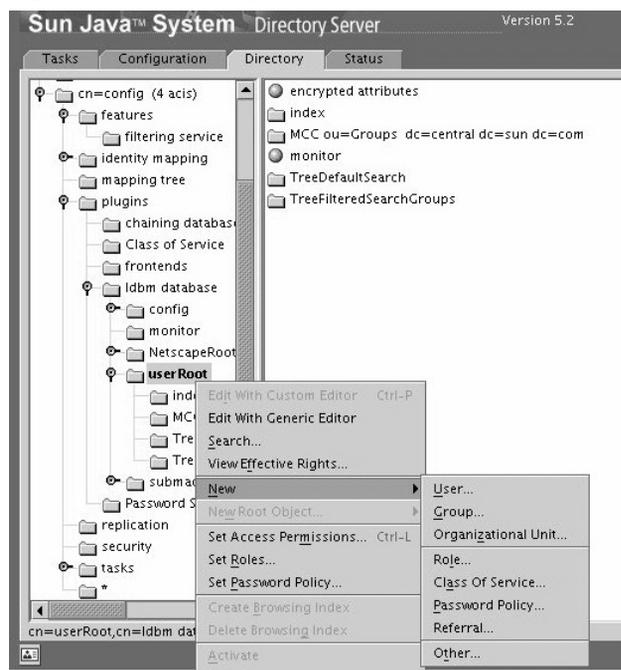
Use the following steps to create an index specifically for a “Virtual List View.”

1. Create a `vlvSearch` object under the following dn:

```
cn={dbname},cn=lbdm database,cn=Plugins,cn=config
```

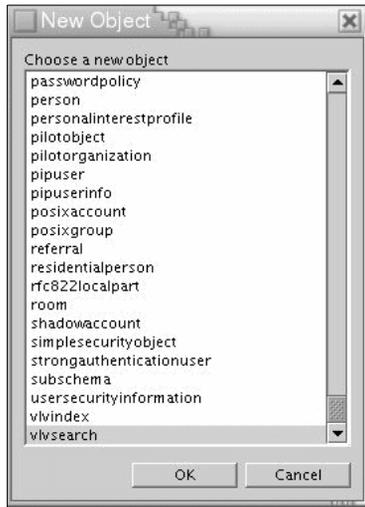
Right-click on the entry in the tree, and when the pop-up menu is displayed, select **New > Other** (Figure C-1).

Figure C-1 Creating a `vlvSearch` Object



2. When the New Object dialog is displayed (Figure C-2), select the `vlvsearch` object class and then click OK.

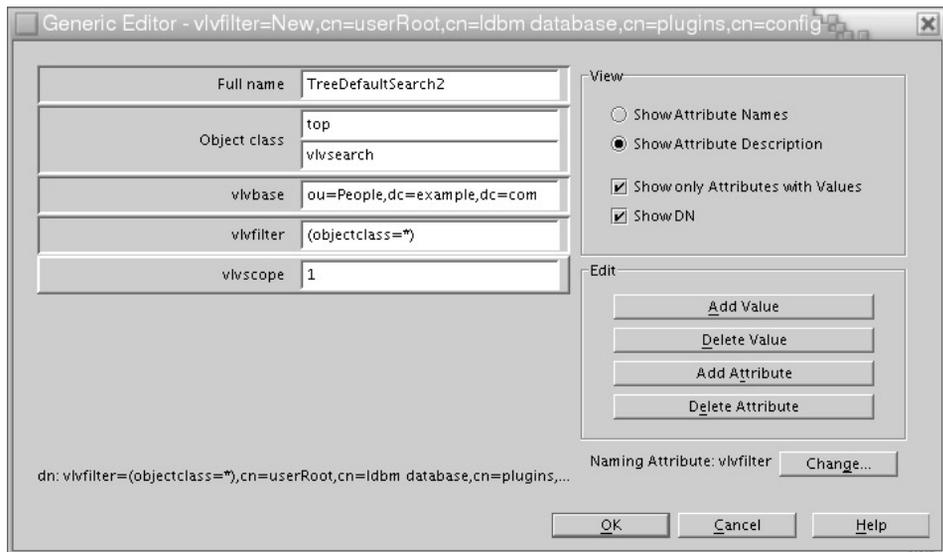
Figure C-2 Selecting the vlvSearch Object Class



The New Object dialog box closes and the Generic Editor dialog box is displayed.

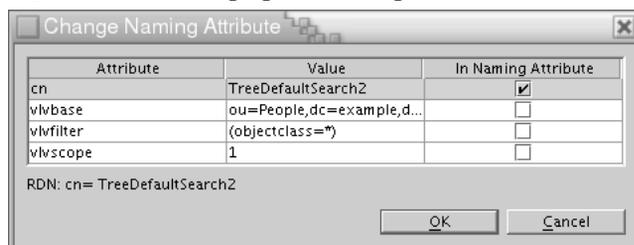
3. Specify the following object class attributes (as shown in Figure C-3):

Figure C-3 Specifying the Object Class Attributes



- **vlvBase:** Specify the base DN used in the query you want to index.
For example:
`ou=People,dc=example,dc=com`
 - **vlvScope:** Specify the LDAP scope of the query you want to index.
The `vlvScope` search scope is similar to an LDAP protocol search in which the scope is a number, as follows:
 - **0** is the entry
 - **1** is only the entries exactly one level below the search base
If you set `vlvScope` to **1**, you must create a `vlvSearch/vlvIndex` for each `ou` where you want a VLV index.
 - **2** is the entry and all descendants
 - **vlvFilter:** Specify the LDAP filter used in the query you want to index.
For example:
`(objectClass=*)`
4. Before closing the Generic Editor dialog box, change the naming attribute to `cn`, which will enable you to provide a more user-friendly name. Click the Change button and the Change Naming Attribute dialog box is displayed (Figure C-4).

Figure C-4 Changing the Naming Attributes

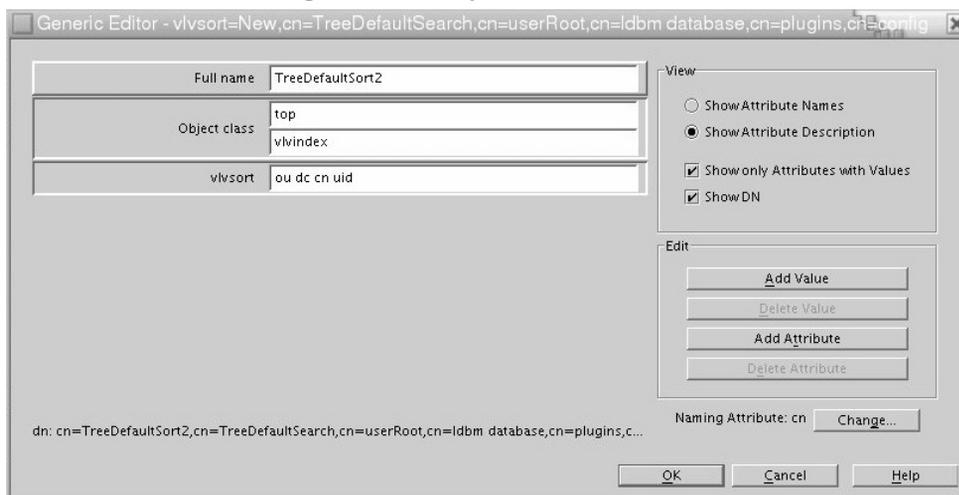


5. When you are finished, click OK to save the new attribute name and close the Change Naming Attribute dialog box.
6. Specify a more user-friendly name for the `cn` attribute.
7. After you have created the `vlvSearch` object, you must create a `vlvIndex` object under the `vlvSearch` object.

The steps for creating a `vlvIndex` object are a very similar to the steps you just used to create the `vlvSearch` object. The only difference is that you will be creating a `vlvIndex` object instead of a `vlvSearch` object, which means the Generic Editor dialog will be slightly different.

8. Specify the `vlvsort` attribute for the sort you will be using for the VLV query. As in the preceding steps, specify a user-friendly name for the object. For example, see Figure C-5.

Figure C-5 Creating a `vlvIndex` Object Class



9. Shut down the directory server by typing:

```
slapd-<instancename>/stop-slapd
```

10. After creating the preceding entries, you must prompt the directory server to create the index as follows:

Use the `vlvindex` command in the `slapd-<instancename>` directory, which is located at the same level as the `start-slapd` command. The `vlvindex` command has two parameters:

- o **-n:** Specifies the database
- o **-T:** Specifies the name of the `vlvIndex` object you are using to create the index

For example:

```
$ ./vlvindex -n userRoot -T "Tree Default Sort 2"
```

From the Command Line

Perform the following steps to create a VLV index:

1. Create an entry with the `vlvSearch` object class below the entry:

```
cn={dbname},cn=lbdm database,cn=plugins,cn=config
```

The `vlvSearch` object defines the search parameters for the index you want to create. You should give the entry a meaningful name and change the default naming attribute that the object will use for this name.

The `vlvSearch` object class requires the following attributes:

- **vlvBase:** Specifies the search base DN to which the index will apply. For example:

```
ou=People,dc=example,dc=com
```

- **vlvScope:** Specifies the scope of the search to be performed.

The `vlvScope` search scope is similar to an LDAP protocol search in which the scope is a number, as follows:

- **0** is a `baseObject` search that includes only the entry specified as the search base.
- **1** is a `singleLevel` search that includes all entries exactly one level below the search base.
- **2** is a `wholeSubtree` search includes the entry specified as the search base and all entries anywhere below it.

- **vlvFilter:** Specifies a filter for the search. For example:

```
(objectClass=*)
```

2. Create an entry under the `vlvSearch` entry.

This new entry requires an `vlvIndex` object class, which defines the sorting to use in your search. You should give the entry a meaningful name and change the default naming attribute that the object will use to this name.

The `vlvIndex` object requires the `vlvSort` attribute, which specifies a list of attributes on which to sort, and in what order to sort them. For example,

```
"ou dc cn uid"
```

- After creating the preceding entries, you must prompt the directory server to create the index as follows:

Use the `vlvIndex` command in the `slapd-<instancename>` directory, which is located at the same level as the `start-slapd` command. The `vlvIndex` command has two parameters:

- n:** Specifies the database
- T:** Specifies the name of the `vlvIndex` object you are using to create the index

For example:

```
$ ./vlvindex -n userRoot -T Tree Default Sort 2
```

Creating Tree Indexes

You can make the browse tree more efficient by creating a virtual list view index. Use the following settings for this index.

- To browse the index when the tree is not filtering anything (assume you are creating an index for `ou=People,dc=example,dc=com`):

Table C-1 Settings When the Tree is Not Filtering

vlvSearch object:	dn: cn=TreeDefaultIndex,cn=UserRoot,cn=lbdm database, cn=Plugins,cn=config cn: TreeDefaultIndex vlvScope: 1 vlvBase: ou=People,dc=example,dc=com vlvFilter: (objectclass=*) objectClass: top objectClass: vlvSearch
vlvIndex object:	dn: cn=TreeDefaultSort,cn=TreeDefaultIndex,cn=UserRoot, cn=lbdm database, cn=Plugins, cn=config cn: TreeDefaultSort vlvsort: ou dc cn uid objectClass: top objectClass: vlvIndex

- To browse the index when the tree is filtering groups (assume the tree is browsing the `ou=People,dc=example,dc=com`):

Table C-2 Settings to When the Tree is Filtering Groups

<code>vlvSearch object</code>	<pre>dn: cn=TreeDefaultIndex,cn=UserRoot,cn=lbdm database, cn=Plugins,cn=config cn: TreeDefaultIndex vlvScope: 1 vlvBase: ou=People,dc=example,dc=com vlvFilter: ((&(hasSubordinates=TRUE)(numSubordinates>=0)) ((objectClass=groupofuniquenames) (objectClass=groupofnames))) objectClass: top objectclass: vlvSearch</pre>
<code>vlvIndex object</code>	<pre>dn: cn=TreeDefaultSort,cn=TreeDefaultIndex, cn=UserRoot, cn=lbdm database,cn=Plugins,cn=config cn: TreeDefaultSort vlvsort: ou dc cn uid objectClass: vlvIndex objectclass: Top</pre>

- To browse the index when the tree is filtering users (assume the tree is browsing the `ou=People,dc=example,dc=com`):

Table C-3 Settings When Tree is Filtering Users

<code>vlvSearch object</code>	<pre>dn: cn=TreeDefaultIndex,cn=UserRoot,cn=lbdm database, cn=Plugins,cn=config cn: TreeDefaultIndex vlvScope: 1 vlvBase: ou=People,dc=example,dc=com vlvFilter: ((&(hasSubordinates=TRUE) (numSubordinates>=0))((objectClass=inetorgperson) (objectClass=organizationalperson)(objectClass=person) (objectClass=inetuser))) objectclass: Top objectclass: vlvSearch</pre>
<code>vlvIndex object</code>	<pre>dn: cn=TreeDefaultSort,cn=TreeDefaultIndex, cn=UserRoot,cn=lbdm database,cn=Plugins,cn=config cn: TreeDefaultSort vlvsort: ou dc cn uid objectclass: Top objectclass: vlvIndex</pre>

- To browse the index when the tree is filtering users and the testuser string (assume the tree is browsing the ou=People,dc=example,dc=com DIT):

NOTE These indexes are deployment-specific, and they are provided here for documentary purposes only. You can decide whether these indexes should be required.

Table C-4 Settings When the Tree is Filtering Users and the testuser String

vlvSearch object:	dn: cn=TreeDefaultIndex,cn=UserRoot,cn=lbdm database, cn=Plugins,cn=config cn: TreeDefaultIndex vlvScope: 1 vlvBase: ou=People,dc=example,dc=com vlvFilter: ((&(hassubordinates=TRUE) (numsubordinates>=0))(&((objectclass=inetorgperson) (objectclass=organizationalperson)(objectclass=person) (objectclass=inetuser))((uid=*testuser*)(cn=*testuser*) (dc=*testuser*)(ou=*testuser*))))) objectClass: top objectclass: vlvSearch
vlvIndex object:	dn: cn=TreeDefaultSort,cn=TreeDefaultIndex,cn=UserRoot, cn=lbdm database,cn=Plugins,cn=config cn: TreeDefaultSort vlvsort: ou dc cn uid objectClass: top objectclass: vlvSearch

- To browse the index when the tree is filtering the testuser string (assume the tree is browsing the ou=People,dc=example,dc=com DIT):

Table C-5 Settings When the Tree is Filtering the testuser String

vlvSearch object:	dn: cn=TreeDefaultIndex,cn=UserRoot,cn=lbdm database, cn=Plugins,cn=config cn: TreeDefaultIndex vlvScope: 1 vlvBase: ou=People,dc=example,dc=com vlvFilter: ((&(hassubordinates=TRUE) (numsubordinates>=0))((uid=*testuser*)(cn=*testuser*) (dc=*testuser*)(ou=*testuser*)))
-------------------	---

Table C-5 Settings When the Tree is Filtering the testuser String

vlvIndex object:	dn: cn=TreeDefaultSort,cn=TreeDefaultIndex,cn=UserRoot, cn=lbdm database,cn=Plugins,cn=config cn: TreeDefaultSort vlvsort: ou dc cn uid objectClass: top objectclass: vlvSearch
------------------	--

Using the Virtual List View Anonymously

By default, a Sun Java System Directory Server deployment does not allow unauthenticated users to use the Virtual List View control.

If you want to permit anonymous users to access the Virtual List View features (to give them increased performance when using Directory Editor Browse or Search pages) then you must configure Directory Server to allow anonymous-user access for VLV searches.

However, it is important to remember that by doing this, you expose Directory Server to resource-intensive requests from an unauthenticated user and you should consider taking steps to protect Directory Server from arbitrary connections. For example, you could put this Directory Server behind a Directory Proxy Server and ensure proper resource limits.

To configure Directory Server to permit anonymous users to use the Virtual List View control, you must change the `aci` attribute of the entry with the following dn:

```
dn: oid=2.16.840.1.113730.3.4.9,cn=features,cn=config
```

By default, the `aci` attribute for this entry is:

```
aci: (targetattr != "aci")(version 3.0; acl "VLV Request Control";  
allow( read, search, compare, proxy ) userdn = "ldap:///all";)
```

This configuration means that any authenticated Directory Server user has access to this entry, which in turn grants access to the Virtual List View control. Consequently, because the anonymous user is not authenticated they are not allowed access.

The `aci` attribute should be changed as follows:

```
aci : (targetattr != "aci")(version 3.0; acl "VLV Request Control"; allow(  
read, search, compare, proxy ) userdn = "ldap:///anyone";)
```

This configuration gives anyone access to the entry — whether they are authenticated or not — and they will have permission to use the Virtual List View control.

Index

A

- access
 - anonymous log-ins 30
 - configuring policy files 14, 18, 177
 - controlling for applications 141
 - Directory Editor permissions 2, 9, 179
 - dynamic groups 72
 - logging directory access 175
 - logging in 38
 - Manager Group 34
 - WEB-INF/classes directory 32
- action attributes 195
- action components 207
- actions and views for role configuration 207
- Advanced Search tab 123
- align property 196
- allowed values property 192
- anonymous log-ins 30
- Apache Tomcat 2
- application servers
 - configuring policy files 14, 18, 177
 - deploying Directory Editor 22, 25
 - deploying Directory Editor on WebLogic 26
 - installing Sun 13
 - installing Tomcat 17
 - installing WebLogic 24
 - Sun 10, 13
 - supported 2
 - undeploying Directory Editor 168
- applications, controlling access 141
- attribute values 98
- attributes
 - action 195
 - adding/removing 35, 52, 125, 135, 136, 153
 - calculating hidden 100
 - creating 153
 - default 152
 - defining values 98, 100
 - deleting 156
 - displaying 100
 - dollar sign (\$) 133
 - editing 86, 131, 153, 155, 181, 208, 209, 213
 - favored 136
 - inetorgperson 98
 - LDAP 110
 - lists 125
 - Locality 77
 - logged information 175
 - Mail User 59
 - managing 217
 - mapping 153
 - member 70, 71
 - memberURL 72
 - Naming Attribute Mapping table 153
 - naming attributes 151, 152
 - Naming Attributes page 151
 - naming corresponding fields 110
 - onClick 195
 - organizationalperson 98
 - organizationalunit 98
 - read-only 100
 - registeredAddress 79, 83
 - removing 156
 - required 51, 61, 65, 76, 80, 84, 181
 - schema map 193

- searching for 125
- selecting 125
- specifying Common Attributes 52, 77, 81
- specifying Default NT User 61
- specifying extensions 57
- specifying for managed directory 35
- specifying Locality 83
- specifying Misc Attributes 56
- specifying UNIX User 63
- uniqueMember 70, 71
- using single quotes 195
- view 191, 192, 201
- audit logs 175
- authenticating
 - clients 141
 - users 34
- authentication, description/definition 141
- Authorization page
 - accessing 142, 144
 - controlling user access 141
 - defining roles 145
 - deleting roles 151
 - editing roles 149
- authorization, description/definition 141
- auxiliary classes
 - selecting 116

B

- BackLink component 197
- base component classes 190
- base context, specifying 34
- Basic Component classes 186
- Basic Search tab 120
- BEA WebLogic 2
- Browse
 - button 44
 - dialog box 44
 - page 30, 38, 41
- browsers, supported 3
- browsing
 - Browse dialog box 44
 - directory contents 41

- with anonymous log-in 30, 38
- build numbers 172
- Button component 198

C

- capabilities 148, 207
 - description 143
- Capabilities tab 148
- central log directories xi
- changing LDAP passwords 39
- Checkbox component 198
- class of service (COS) template 98
- classes
 - adding to enterprise 153
 - adding to forms 116
 - Basic Component 186
 - Component 186, 190, 192
 - configuring logging levels 174
 - container 186, 187, 188
 - custom 138
 - datastore.dmlIdAttribute 31
 - datastore.location 31
 - datastore.objectClass 31
 - datastore.xmlObjectAttribute 31
 - default naming attributes 152
 - editing 181
 - groupOfNames 65
 - groupOfUniqueNames 65
 - groupOfURLs 72
 - HTML display components 186
 - Label 204
 - List 189
 - location 181
 - mapping naming attributes 153
 - removing 88
 - selecting auxiliary 116
 - selecting object 124
 - selecting structural 115
 - specifying naming attributes 154
 - string 189
- clients, authenticating 141
- colspan property 196

- command property 194
- commands
 - Directory Editor conventions xi
 - extracting de.war 178
 - installing Directory Editor 10
 - java 4, 10, 165
 - launching uninstall wizard 165
 - minimum/maximum heap size 4
 - starting Directory Editor 22, 24
 - stopping Directory Editor 23, 24
 - updating de.war 178
- comments and suggestions xiv
- Common Attributes, specifying 52, 77, 81
- Component classes 186, 190, 192
- components
 - BackLink 197
 - Button 198
 - Checkbox 198
 - classes 186
 - DatePicker 199
 - Javascript 199
 - Label 200
 - Link 201
 - MultiSelect 201
 - Radio 203
 - SectionHead 204
 - Select 204
 - SimpleTable 205
 - subclasses 188
 - Text 205
 - Text Area 206
- configuration directory 5, 29, 30, 42, 181
- Configuration Directory Server parameters 29
- configuring
 - default search filters 132, 133
 - Directory Editor 41
 - policy files 2, 14, 18, 177
 - Results table 134
 - Search page 119, 131
 - search types 137
 - servers 3
 - WebLogic server 25, 26
- contacting tech support 171
- container classes 186, 187, 188
- controlling access
 - applications/application components 141
- conventions
 - default paths and file names xi
 - mnemonics xi
 - naming 188
 - naming attributes 152
 - symbol xi
 - typographic x
- Create page
 - accessing 41, 50, 122
 - Common Attributes tab 52, 81, 84
 - creating domains 80
 - creating groups 64
 - creating new users 50
 - Default NT User tab 61
 - Group Membership tab 54
 - Import/Export tab 71
 - Locality tab 81
 - Location tab 55
 - Misc Attributes tab 56
 - Other tab 82
 - Quick Create tab 83
 - Static Group tab 66
 - Unix User tab 63
- creating
 - domains 80
 - form fields 114
 - forms 115
 - groups 64
 - localities 83
 - naming attributes 153
 - objects 49
 - organizations 75
 - roles 145
 - users 50
- credentials 31, 38, 40, 141, 215
- custom classes 138
- customizing
 - default fields 100
 - default forms 99
 - field locations 112
 - form field properties 109
 - forms 99, 103

D

- data types 189
- database directories, supported 3
- datastore.dmlIdAttribute 31, 181
- datastore.dmlIdAttribute property 182
- datastore.location 31
- datastore.location property 181
- datastore.objectClass 31
- datastore.objectClass property 181
- datastore.xmlObject Attribute 31
- datastore.xmlObjectAttribute 181
- datastore.xmlObjectAttribute property 182
- DatePicker component 199
- de.home, defining 14
- de.war file 23
 - default installation directory xi
 - extracting 178
 - installing 24, 26
 - installing manually 23
 - locating 22
 - removing 166, 167
 - updating 178
- default
 - application server instance folders 14
 - BackLink value 197
 - Button value 198
 - changing default search types 137
 - changing limits 119, 126, 130
 - configuration information location 167, 168, 181
 - configuring default search filter 132
 - fields 100
 - filtered search string 129
 - forms 97
 - group object 65
 - inetorgperson 138
 - installation directory xi
 - log files 174
 - log4j.properties file 174
 - naming attributes 152
 - noNewRow value 193
 - padding 200, 204
 - paths and filenames xi
 - ports 17
 - roles 142, 151
 - search filters 133
 - search method 41
 - search results limits 119, 126, 130
 - setting default values 99
 - user extensions 57
- Default Create Form 103
- Default NT User attributes, specifying 61
- Default NT User tab 61
- Default role 142, 144, 150
- defining 98
 - attribute values 100
 - de.home variables 14
 - domains 98
 - roles 145
- Delete page, accessing 122
- deleting
 - classes 88
 - form fields 114
 - naming attributes 153, 156
 - objects 95
 - roles 151
- deploying Directory Editor 16, 22, 25, 26
- directories
 - browsing 41
 - central log default xi
 - creating objects 41
 - default paths and filenames xi
 - Directory Editor installation 11
 - editing information 41
 - managed 33
 - searching 41
 - top-level objects 42
 - WEB-INF/classes 21, 32
- Directory Editor
 - configuring 41
 - installing 10, 13
 - installing manually 23
 - installing on WebLogic 24
 - installing on WebSphere 26
 - installing with Tomcat Manager 22
 - navigating 40
 - starting 22, 24
 - stopping 23, 24
 - undeploying 168
 - uninstalling 165
- directory information 41

- Directory Server
 - authenticating to 29
 - installation prerequisites 4
 - logs 175
 - port number 29
 - publications viii
- Disable page
 - accessing 122
- disabling
 - objects 92
- DisplayClass element 186
- DNs
 - downloading 89, 210
 - exporting 98
 - invalid 70, 71
- documentation
 - downloading xii, 13
 - overview xii
 - recommended
 - reading xii
- dollar sign (\$) 133
- domains
 - creating 80
 - defining 98
 - mail user attributes 59
 - NT user attributes 61
 - selecting 24
 - Unix user attributes 63
- downloading
 - J2SDK 8
 - JCE 1.2.2 180
 - member DNs 89
 - product documentation xii, 13
 - Sun products xii
- Dynamic Group tab 72
- dynamic groups
 - access control 72
 - adding filters 73

E

- Edit an Object page
 - accessing 41, 88, 122

- Edit page
 - accessing 44, 86
- Edit Password page, accessing 41
- Edit Roles page 150
- editing 41
 - attributes 86, 131, 155, 181, 208, 209, 213
 - filters 75
 - LDAP passwords 41
 - Managed Directory page 157
 - naming attributes 155
 - objects 86
 - roles 149
 - Search page 119, 131
 - Startup Properties page 157
 - startup.properties 178
- editing objects 49
- Enable page
 - accessing 122
- enabling
 - objects 92
- exceptions
 - policy files 9, 179, 180
 - startup.properties 178
- exporting group members 89
- extensions
 - specifying 57
 - user 57

F

- favored Attributes 136
- features 37
- field locations, customizing 112
- field properties
 - customizing 109
 - descriptions 110
- Field Properties tab 112, 114
- fields
 - creating 114
 - customizing 100
 - deleting 114
- files
 - installed with Directory Editor 10

- filtered searches 129
 - default search string 129
- filters
 - adding to dynamic groups 73
 - editing 75
 - removing 75
- Firefox 3
- forms
 - adding classes 116
 - changing field locations 112
 - creating 115
 - creating fields 114
 - customizing 99, 103
 - customizing field properties 109
 - default 97
 - Default Create Form 103
 - deleting fields 114
 - descriptions 97
 - field properties 106, 110
 - identification 107
 - move fields 106
 - opening 103
 - overview 99
 - property view 103, 105
 - setting default values 99
 - standard/default 98
 - testing 118
 - tree view 103, 104
- Forms page
 - accessing 101
 - description 102

G

- groupOfNames 65
- groupOfUniqueNames 65
- groupOfURLs 72
- groups
 - adding members 67
 - assigning users 54
 - creating 64
 - default object 65
 - importing/exporting members 89
 - removing members 69

- searching for 120
- showing members 71
- viewing members 69, 71

H

- heap size 4
- Help 42, 46
- help property 194
- hidden attributes 100
- hidden parameters 187
- Home page 39
- hosts
 - adding/removing 33
- HTML display components 186

I

- IBM WebSphere 2
- identifying forms 107
- Import/Export Tab 71
- importing group members 89
- inetorgperson 98, 138
- installation instructions 7
- installer, launching 10
- installing
 - application servers 8
 - Directory Editor 10, 13
 - Directory Editor for Sun application servers 13
 - Directory Editor for Tomcat 17
 - Directory Editor for WebLogic 24
 - Directory Editor for WebSphere 26
 - Directory Editor manually 23
 - specifying the directory and path 11
 - status 11
 - Sun application servers 13
 - Tomcat application servers 17

J

J2SDK

- downloading 8
- preferred 8
- requirements 4, 7

Java classes 186

java commands

- adding heap size 4
- installing Directory Editor 10
- uninstall 165

Java compilers 8

Java Development Kit. *See J2SDK.*Java Virtual Machine. *See JVM.*

JAVA_HOME 18

- adding 9
- setting 18

Javascript component 199

JCE, downloading 180

JVM

- adding options for de.home 14
- increasing memory 177
- properties to allow read access 20
- running Java classes 8
- setting heap size 4

List classes 189

localities, creating 83

Locality attributes, specifying 77, 83

location property 194

locations, user 55

Log In page 38

log4j.properties file 174

logging

- attribute changes 175
- auditing messages 175
- configuring logging levels 174
- setting levels 173

logging in

- anonymously 30
- Directory Editor Log In page 38
- error messages 31

logging out 42

logs

- application server 176
- default 174
- default path xi
- default paths and file names xi
- Directory Server 175
- getting information 173
- java.security.AccessControlException 179
- WSTrace.log file 174

K

key icon 105

L

Label class 204

Label component 200

launching the installer 10

LDAP attributes 110

LDAP passwords

- changing 39

LDAP passwords, editing 41

license agreement 10

Link component 201

M

mail accounts 59

Mail User attributes, specifying 59

managed directory 5

- specifying attributes 35
- specifying properties 33

Managed Directory page

- editing 157
- initial configuration 33

manager principals 34, 157

Manager role 22, 142, 143, 150

managing attributes 217

mapping naming attributes 153

member attribute 70, 71

- member DNs, downloading 89
- memberURL attribute 72
- memory
 - increasing 177
 - requirements 4
- Microsoft Internet Explorer 3
- Misc Attributes, specifying 56
- mnemonics, using xi
- Move Field tab 112
- Mozilla 3
- MultiSelect component 201

N

- name property 191
- naming attributes
 - accessing Naming Attributes page 151
 - creating 153
 - default 152
 - deleting 156
 - editing 155
 - mapping 153
 - Naming Attribute Mapping table 153
 - removing 156
 - specifying 35
- Naming Attributes page 152
- naming conventions 188
- navigation tool 136
- Netscape 3
- New Role page 145
- node keys 43
- noNewRow property 193
- non-production mode 177
- nowrap property 196

O

- object classes
 - adding 153
 - default naming attributes 152

- editing 181
- location 181
- mapping naming attributes 153
- specifying naming attributes 154
- objects
 - creating 49
 - creating domains 80
 - creating groups 64
 - creating localities 83
 - creating organizations 75
 - creating users 49
 - deleting 95
 - editing 49
 - enabling/disabling 92
 - renaming 93
 - searching for 123
 - top-level directory 42
- onChange property 196
- onClick attribute 195
- onClick property 195
- online Help 42, 46
- online resources xii
- online support xii
- operating systems, supported 2
- organizational units
 - searching for 120
- organizationalperson attribute 98
- organizationalunit attribute 98
- organizations
 - creating 75

P

- page source
 - viewing 172
- pages
 - Edit 44
 - Home 39
 - Quick Create 65
 - Search Configuration 132
 - Search for Principals 146, 158
- parameters
 - hidden 187

- passwords
 - changing 39
 - editing 41
- patches
 - information about xii
- performance
 - impacts 28, 133, 180
 - optimizing 3, 4
- permissions 2
- policy files
 - configuring application servers 2
 - configuring Sun Application Server 14, 177
 - configuring Tomcat 18
 - exceptions 9, 179, 180
- ports, specifying 29
- prerequisites
 - recommended reading viii
- primary key property 193
- principals
 - adding/removing 34, 157
 - description/definition 142, 143
 - representing with directory objects 143
 - searching for 157
 - specifying 146
- product documentation
 - downloading xii, 13
 - related xii
- product downloads xii
- product features 37
- product support xiii
- properties
 - align 196
 - allowed value 192
 - colspan 196
 - command 194
 - datastore.dmlIdAttribute 182
 - datastore.location 181
 - datastore.objectClass 181
 - datastore.xmlObjectAttribute 182
 - help 194
 - location 194
 - managed directory 33
 - name 191
 - naming conventions 188
 - noNewRow 193

- nowrap 196
- onChange 196
- onClick 195
- primary key 193
- required 193
- title 191
- value 192
- width 196
- Property elements 186
- property view 103, 105
- publications
 - Directory Server viii, xii
 - downloading xii
 - related xii

Q

- query strings 129
- Quick Create page 65, 80
- quotes, using 195

R

- Radio component 203
- read-only attributes 100
- recommended
 - reading viii, xii
 - server configurations 3
- registeredAddress attribute 79, 83
- related documentation xii
- removing
 - de.war file 166, 167
 - Directory Editor 165
 - filters 75
 - naming attributes 153, 156
- Rename page
 - accessing 122
- renaming objects 93
- required
 - attributes 51, 61, 65, 76, 80, 84, 181
- required property 193

- requirements
 - application servers 2
 - browsers 3
 - database directories 3
 - J2SDK 4, 7
 - Java compilers 8
 - memory 4
 - operating systems 2
- resources
 - online xii
- Restore Default Setting button 143, 144, 150
- Results table
 - configuring 134
 - using 122
- roles
 - capabilities 207
 - default 142, 151
 - defining 145
 - deleting 151
 - description/definition 142
 - editing 149

S

- schema map attributes 193
- Search Configuration page
 - accessing 132
 - configuring default search filters 133
 - configuring Results table 135
 - configuring search types 137
- Search for Principals page 146, 158
- search methods
 - advanced 123
 - authenticating users 34
 - basic 120
 - descriptions 41
 - example searches 126
 - filtered 129
- Search page
 - accessing 41
 - Advanced Search tab 123
 - Basic Search tab 120
 - configuring 119, 131
 - configuring default search filters 132, 133
 - editing 119, 131
 - Filtered Search tab 129
 - search types 119
- search results
 - changing default limits 119, 126, 130
 - maximum results 120, 130
- search types
 - changing default 137
 - configuring 137
- searching
 - advanced search 123
 - basic search 120
 - default filtered search string 129
 - examples 126
 - filtered search 129
 - for attributes 125
 - search conditions 125
- searching for manager principals 157
- SectionHead component 204
- Select component 204
- server configurations, recommended 3
- setclasspath.bat file 18
- setclasspath.sh file 18
- SimpleTable component 205
- single quotes, using 195
- software, supported 1
- specifying
 - base context 34
 - Managed Directory page properties 33
 - principals 146, 157
 - Startup Properties page properties 28
 - startup.properties 28
- starting Directory Editor 22, 24
- Startup Properties page
 - editing 157
 - initial configuration 28
- startup.properties
 - exceptions 178
 - persisting 32
 - saving 179
 - specifying 28, 30, 31, 178
- statistics, viewing 44
- stopping and restarting
 - Tomcat 23
 - WebLogic 24

- stopping Directory Editor 23, 24
- string classes 189
- structural classes
 - selecting 115
- subclasses, component 188
- subject
 - description/definition 143
- suggestions and comments xiv
- Sun application servers
 - installing 10, 13
 - installing Directory Editor 13
 - undeploying Directory Editor 168
- Sun Java™ System Application Server 2
- Sun Java™ System Directory Server
 - installation prerequisites 4
- Sun ONE Application Server 2
- Sun online resources xii
- support, contacting 171
- support, product xii, xiii
- supported
 - application servers 2
 - browsers 3
 - database directories 3
 - operating systems 2
 - software 1
- symbol conventions xi

T

- tabs
 - Advanced Search 123
 - Basic Search 120
 - Capabilities 148
 - Common Attributes 52, 81, 84
 - Field Properties 112, 114
 - Filtered Search 129
 - Group Membership 54
 - Import/Export tab 71
 - Locality 81
 - Location 55
 - Misc Attributes 56
 - Move Field 112
 - Other 82

- Quick Create 83
- Static Group 66
- Unix User 63
- technical support xiii
- technical support, contacting 171
- testing forms 118
- Text Area component 206
- Text component 205
- third-party websites xiii
- title property 191
- Tomcat application servers
 - deploying Directory Editor 22
 - installing 17
 - installing Directory Editor 17
 - policy file 18
 - stopping and restarting 23
 - undeploying Directory Editor 169
- Tomcat Manager, installing Directory Editor 22
- tree view 103, 104
- typographic conventions x

U

- undeploying Directory Editor 168
- uninstall wizard 165
- uninstalling Directory Editor 165
- uniqueMember attribute 70, 71
- UNIX User attributes, specifying 63
- updating de.war 178
- user view attributes 99
- users
 - assigning to groups 54
 - authenticating 34
 - creating 50
 - default extensions 57
 - mail accounts 59
 - miscellaneous information 56
 - physical location 55
 - searching for 120
 - Windows NT accounts 61

V

- value property 192
- variables
 - adding JAVA_HOME 9
 - de.home 14
- view attribute 191, 192, 201
- view components 207
- viewing
 - build number 172
 - directory contents 41
 - folder contents 43
 - form elements 104
 - group members 69, 71
 - page source 172
 - statistics 44

W

- WEB-INF/classes directory permissions 21, 32
- WebLogic
 - configuring server 25, 26
 - deploying Directory Editor 25, 26
 - installing Directory Editor 24
 - stopping and restarting 24
 - undeploying Directory Editor 169
 - Web application home directory 24
- websites
 - comments and suggestions xiv
 - Directory Server publications viii, xii
 - Microsoft product documentation xiii
 - Sun product documentation vii
 - Sun resources xii
 - support xiii
 - third-party xiii
- WebSphere
 - installing Directory Editor 26
 - undeploying Directory Editor 170
- width property 196
- WSTrace.log file 174