First Edition (March 2001)

This edition, SC24-5909-00, applies to Version 1 Release 1 of z/OS DCE Base Services, z/OS DCE user Data Privacy (DES and CDMF), z/OS DCE User Data Privacy (CDMF) (program number 5694-A01), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for reader's comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Information Development, Dept. G60
1701 North Street
Endicott, NY 13760-5553
United States of America

FAX (United States & Canada): 1+607+752-2327
FAX (Other Countries):
Your International Access Code +1+607+752-2327

IBMLink™ (United States customers only): GDLVME(PUBRCF)
Internet e-mail: pubrcf@vnet.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1994, 2001. All rights reserved.
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
The following statements are provided by the Open Software Foundation.

The information contained within this document is subject to change without notice.

OSF MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

OSF shall not be liable for errors contained herein, or for any direct or indirect, incidental, special or consequential damages in
connection with the furnishing, performance, or use of this material.

Copyright © 1993, 1994 Open Software Foundation, Inc.

This documentation and the software to which it relates are derived in part from materials supplied by the following:

- © Copyright 1990, 1991 Digital Equipment Corporation
- © Copyright 1990, 1991 Hewlett-Packard Company
- © Copyright 1989, 1990, 1991 Transarc Corporation
- © Copyright 1990, 1991 Siemens Nixdorf Informationssysteme AG
- © Copyright 1990, 1991 International Business Machines Corporation
- © Copyright 1988, 1989 Massachusetts Institute of Technology
- © Copyright 1988, 1989 The Regents of the University of California

All Rights Reserved.

Printed in the U.S.A.

THIS DOCUMENT AND THE SOFTWARE DESCRIBED HEREIN ARE FURNISHED UNDER A LICENSE, AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. TITLE
TO AND OWNERSHIP OF THE DOCUMENT AND SOFTWARE REMAIN WITH OSF OR ITS LICENSORS.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software
Foundation, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

DEC, DIGITAL, and ULTRIX are registered trademarks of Digital Equipment Corporation.

DECstation 3100 and DECnet are trademarks of Digital Equipment Corporation.

HP, Hewlett-Packard, and LaserJet are trademarks of Hewlett-Packard Company.

Network Computing System and PasswdEtc are registered trademarks of Hewlett-Packard Company.

AFS and Transarc are registered trademarks of the Transarc Corporation.

Episode is a trademark of the Transarc Corporation.

Ethernet is a registered trademark of Xerox Corporation.

DIR-X is a trademark of Siemens Nixdorf Informationssysteme AG.

MX300i is a trademark of Siemens Nixdorf Informationssysteme AG.

NFS, Network File System, SunOS and Sun Microsystems are trademarks of Sun Microsystems, Inc.

X/Open is a trademark of The Open Group in the U.K. and other countries.

PostScript is a trademark of Adobe Systems Incorporated.

FOR U.S. GOVERNMENT CUSTOMERS REGARDING THIS DOCUMENTATION AND THE ASSOCIATED SOFTWARE

These notices shall be marked on any reproduction of this data, in whole or in part.

NOTICE: Notwithstanding any other lease or license that may pertain to, or accompany the delivery of, this computer software,
the rights of the Government regarding its use, reproduction and disclosure are as set forth in Section 52.227-19 of the FARS
Computer Software-Restricted Rights clause.

RESTRICTED RIGHTS NOTICE: Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in
subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

RESTRICTED RIGHTS LEGEND: Use, duplication or disclosure by the Government is subject to restrictions as set forth in
paragraph (b)(3)(B) of the rights in Technical Data and Computer Software clause in DAR 7-104.9(a). This computer software is
submitted with “restricted rights.” Use, duplication or disclosure is subject to the restrictions as set forth in NASA FAR SUP
18-52.227-79 (April 1985) "Commercial Computer Software-Restricted Rights (April 1985)." If the contract contains the Clause at
18-52.227-74 “Rights in Data General” then the "Alternate III" clause applies.

US Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract.

Unpublished—All rights reserved under the Copyright Laws of the United States.

This notice shall be marked on any reproduction of this data, in whole or in part.
## Contents

**About This Book** .................................................. xvii
Who Should Use This Book ............................................. xvii
How This Book Is Organized ........................................... xvii
Unsupported Services and Commands ..................................... xviii
Conventions Used in This Book ......................................... xviii
Where to Find More Information .......................................... xix
  Softcopy Publications ............................................. xix
  Internet Sources .................................................. xix
  Using LookAt to Look up Message Explanations ...................... xix
  Accessing Licensed Books on the Web ................................ xx
Product Name ...................................................... xxi

### Chapter 1. DCE and z/OS Commands

dceconf .......................................................... 2
modify DCEKERN daemon .......................................... 3
modify DCEKERN debug ............................................ 7
modify DCEKERN clock ............................................ 9

### Chapter 2. DCE Administration Commands

account .......................................................... 13
account catalog .................................................. 17
account create .................................................. 19
account delete .................................................. 21
account generate ............................................... 22
account help ................................................... 23
account modify .................................................. 24
account operations .............................................. 26
account show ................................................... 27
acl .............................................................. 29
acl check ....................................................... 35
acl delete ....................................................... 36
acl help ......................................................... 37
acl modify ...................................................... 38
acl operations .................................................. 40
acl permissions .................................................. 41
acl replace ..................................................... 42
acl show ......................................................... 44
attrlist ......................................................... 46
attrlist add ..................................................... 47
attrlist getvalues ............................................... 48
attrlist help ................................................... 49
attrlist list ..................................................... 50
attrlist modify .................................................. 51
attrlist operations ............................................... 52
attrlist remove ................................................... 53
aud ............................................................. 54
aud disable ...................................................... 56
aud enable ....................................................... 57
aud help ......................................................... 58
aud modify ....................................................... 59

© Copyright IBM Corp. 1994, 2001
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>aud operations</td>
<td>60</td>
</tr>
<tr>
<td>aud rewind</td>
<td>61</td>
</tr>
<tr>
<td>aud show</td>
<td>62</td>
</tr>
<tr>
<td>aud stop</td>
<td>63</td>
</tr>
<tr>
<td>audevents</td>
<td>64</td>
</tr>
<tr>
<td>audevents catalog</td>
<td>65</td>
</tr>
<tr>
<td>audevents help</td>
<td>66</td>
</tr>
<tr>
<td>audevents operations</td>
<td>67</td>
</tr>
<tr>
<td>audevents show</td>
<td>68</td>
</tr>
<tr>
<td>audfilter</td>
<td>69</td>
</tr>
<tr>
<td>audfilter catalog</td>
<td>71</td>
</tr>
<tr>
<td>audfilter create</td>
<td>72</td>
</tr>
<tr>
<td>audfilter delete</td>
<td>73</td>
</tr>
<tr>
<td>audfilter help</td>
<td>74</td>
</tr>
<tr>
<td>audfilter modify</td>
<td>75</td>
</tr>
<tr>
<td>audfilter operations</td>
<td>76</td>
</tr>
<tr>
<td>audfilter show</td>
<td>77</td>
</tr>
<tr>
<td>audtrail</td>
<td>78</td>
</tr>
<tr>
<td>audtrail help</td>
<td>79</td>
</tr>
<tr>
<td>audtrail operations</td>
<td>80</td>
</tr>
<tr>
<td>audtrail show</td>
<td>81</td>
</tr>
<tr>
<td>cdscache</td>
<td>82</td>
</tr>
<tr>
<td>cdscache create</td>
<td>83</td>
</tr>
<tr>
<td>cdscache delete</td>
<td>85</td>
</tr>
<tr>
<td>cdscache dump</td>
<td>86</td>
</tr>
<tr>
<td>cdscache help</td>
<td>87</td>
</tr>
<tr>
<td>cdscache operations</td>
<td>88</td>
</tr>
<tr>
<td>cdscache show</td>
<td>89</td>
</tr>
<tr>
<td>cell</td>
<td>91</td>
</tr>
<tr>
<td>cell help</td>
<td>92</td>
</tr>
<tr>
<td>cell operations</td>
<td>93</td>
</tr>
<tr>
<td>cell ping</td>
<td>94</td>
</tr>
<tr>
<td>cell show</td>
<td>96</td>
</tr>
<tr>
<td>clearinghouse</td>
<td>98</td>
</tr>
<tr>
<td>clearinghouse catalog</td>
<td>101</td>
</tr>
<tr>
<td>clearinghouse create</td>
<td>102</td>
</tr>
<tr>
<td>clearinghouse delete</td>
<td>103</td>
</tr>
<tr>
<td>clearinghouse disable</td>
<td>104</td>
</tr>
<tr>
<td>clearinghouse help</td>
<td>105</td>
</tr>
<tr>
<td>clearinghouse initiate</td>
<td>106</td>
</tr>
<tr>
<td>clearinghouse operations</td>
<td>107</td>
</tr>
<tr>
<td>clearinghouse repair</td>
<td>108</td>
</tr>
<tr>
<td>clearinghouse show</td>
<td>109</td>
</tr>
<tr>
<td>clearinghouse verify</td>
<td>111</td>
</tr>
<tr>
<td>clock</td>
<td>112</td>
</tr>
<tr>
<td>clock compare</td>
<td>113</td>
</tr>
<tr>
<td>clock help</td>
<td>114</td>
</tr>
<tr>
<td>clock operations</td>
<td>115</td>
</tr>
<tr>
<td>clock set</td>
<td>116</td>
</tr>
<tr>
<td>clock show</td>
<td>117</td>
</tr>
<tr>
<td>clock synchronize</td>
<td>118</td>
</tr>
<tr>
<td>dcecp</td>
<td>119</td>
</tr>
<tr>
<td>Starting and Stopping DCECP</td>
<td>120</td>
</tr>
<tr>
<td>Startup Scripts</td>
<td>121</td>
</tr>
</tbody>
</table>
Environment Variables .......................................................... 121
Administration Objects ....................................................... 122
Common Operations ............................................................. 122
Miscellaneous dcecp Commands ........................................... 124
echo ........................................................................... 124
errtxt ........................................................................... 125
login ........................................................................... 125
logout ........................................................................... 126
quit .............................................................................. 126
resolve ......................................................................... 126
shell ........................................................................... 126
Tcl Notes ......................................................................... 127
Command Processing ............................................................. 128
Running Operating System Commands from DCECP ................. 128
Abbreviations ..................................................................... 129
Syntax ............................................................................. 130
Argument Lists .................................................................. 130
Attribute Lists .................................................................. 131
Attribute Options ............................................................. 131
Lists of Lists .................................................................... 131
Specifying the Binding Information ....................................... 132
Convenience Variables ....................................................... 132
Error Handling and Debugging .......................................... 133
Help .............................................................................. 134
Command History and Command Line Recall ....................... 136
directory ......................................................................... 139
directory add ..................................................................... 142
directory create ................................................................. 143
directory delete ................................................................ 145
directory help ................................................................... 147
directory list .................................................................... 148
directory merge ................................................................. 149
directory modify ................................................................ 151
directory operations .......................................................... 153
directory remove ................................................................ 154
directory show .................................................................. 155
directory synchronize ........................................................ 157
dts ................................................................................. 158
DTS Server Attributes ........................................................ 161
General Counters .............................................................. 162
DTS Server Counters ........................................................ 162
dts activate ...................................................................... 164
dts catalog ....................................................................... 165
dts configure ................................................................... 166
dts deactivate ................................................................... 167
dts help .......................................................................... 168
dts modify ....................................................................... 169
dts operations ................................................................... 170
dts show .......................................................................... 171
dts stop .......................................................................... 173
dts synchronize ................................................................ 174
endpoint ......................................................................... 174
endpoint create .................................................................. 178
endpoint delete .................................................................. 180
endpoint wlb_activate ................................................................. 186
endpoint wlb_off ................................................................. 190
endpoint wlb_on ................................................................. 192
endpoint wlb_show ............................................................... 194
endpoint wlb_ae_show ............................................................. 188
endpoint wlb_show ............................................................... 194
host ................................................................................. 196
host add .......................................................... 199
host catalog ......................................................... 200
host create ........................................................ 201
host delete ...................................................... 203
host help .......................................................... 204
host list .............................................................. 205
host modify .......................................................... 206
host operations ......................................................... 208
host remove ...................................................... 209
host rename ...................................................... 210
host show .......................................................... 211
host start .......................................................... 212
host stop ........................................................... 214
host ping ............................................................ 215
host data ............................................................. 216
host data catalog ..................................................... 220
host data create ......................................................... 222
host data delete ......................................................... 223
host data help ......................................................... 224
host data modify ......................................................... 225
host data operations ........................................................ 228
host data show .......................................................... 229
keytab ............................................................... 230
keytab add .......................................................... 232
keytab catalog ......................................................... 234
keytab create .......................................................... 235
keytab delete .......................................................... 237
keytab help .......................................................... 238
keytab list ............................................................ 239
keytab operations ......................................................... 240
keytab remove ......................................................... 241
keytab show .......................................................... 242
link ................................................................. 243
link create .......................................................... 245
link delete .......................................................... 247
link help ............................................................ 248
link modify .......................................................... 249
link operations ......................................................... 250
link show ............................................................ 251
log ................................................................. 252
Chapter 4. Cell Directory Service

Cell Directory Service Daemons .................................................. 508
Cell Directory Service Administrative Utilities ......................... 509
ldap_addcell ............................................................................ 511
mkdceregister ........................................................................ 513
add directory ........................................................................... 515
add object ............................................................................... 517
cdscp ....................................................................................... 518
  CDS Control Program Commands .............................................. 518
  clear cached server ................................................................ 531
clear clearinghouse .................................................................. 532
create child .............................................................................. 533
create clearinghouse ............................................................... 534
create directory ....................................................................... 535
create link ............................................................................... 536
create object ........................................................................... 538
create replica .......................................................................... 539
define cached server ............................................................. 540
delete child .............................................................................. 542
delete clearinghouse .............................................................. 543
delete directory ....................................................................... 544
delete link ............................................................................... 545
delete object ........................................................................... 546
delete replica .......................................................................... 547
disable clerk ............................................................................. 548
disable server .......................................................................... 549
do .............................................................................................. 550
dump clerk cache ..................................................................... 551
help ............................................................................................ 552
list child .................................................................................... 555
list clearinghouse ..................................................................... 556
list directory ............................................................................. 558
list link ....................................................................................... 560
list object .................................................................................. 562
remove directory ...................................................................... 564
remove link ............................................................................. 566
remove object .......................................................................... 567
set cdscp confidence .............................................................. 569
set cdscp preferred clearinghouse ........................................... 570
set directory ............................................................................. 571
set directory to new epoch ..................................................... 573
set directory to skulk ............................................................. 575
set link ...................................................................................... 576
set object .................................................................................. 578
show cached clearinghouse .................................................... 580
show cached server .................................................................. 582
show cdscp confidence .......................................................... 583
show cdscp preferred clearinghouse ........................................ 584
show cdscp syntax ...................................................... 585
show cell ................................................................. 588
show child ................................................................. 590
show clearinghouse .................................................... 592
show clerk ................................................................. 595
show directory .......................................................... 597
show link ................................................................. 600
show object ............................................................... 602
show replica ............................................................... 604
show server ............................................................... 607

Chapter 5. Distributed Time Service .................................. 609
advertise ................................................................. 610
change ................................................................. 611
create ................................................................. 612
delete ................................................................. 613
disable ................................................................. 614
DTS Control Program .................................................. 615
dtscp ................................................................. 616
   DTS Control Program Commands .................................. 616
enable ................................................................. 619
exit ................................................................. 620
help ................................................................. 621
quit ................................................................. 622
set ................................................................. 623
show ................................................................. 627
synchronize ............................................................ 635
unadvertise ............................................................ 636
update ................................................................. 637

Chapter 6. Security Service .................................................. 639
acledit ............................................................... 641
dcelogin ............................................................... 648
dkdestroy ............................................................. 650
kinit ................................................................. 652
klist ................................................................. 654
loginact ............................................................... 656
mvsexpt ............................................................... 659
mvsimpt ............................................................... 661
pwd_strengthd ........................................................ 663
rgyedit ............................................................... 664
secadmin .............................................................. 676
   The Default Replica and Default Cell .................................... 677
   Automatic Binding to the Master .......................................... 678
   Starting secadmin ....................................................... 678
   Subcommands .......................................................... 679
sec_create_db ......................................................... 685
secd ................................................................. 688
sec_export_db ........................................................ 690
sec_import_db ........................................................ 691
storepw ............................................................... 693

Appendix A. Notices ....................................................... 695
Trademarks ............................................................... 696
Tables

1. DCECP Administration Objects ........................................ 11
2. Alternate DCECP Operation Names ..................................... 130
3. Versions for ENDPOINT ............................................. 176
4. Versions for RPCENTRY ............................................. 344
5. Versions for RPCPROFILE .......................................... 372
6. RPC Control Program Commands .................................... 451
7. IDL Appended Characters ............................................ 459
8. Short and Long File Names ......................................... 460
9. RPCCP Commands .................................................. 472
10. DCECP Equivalent Commands for RPCCP Commands ................. 473
11. Scope of RPC Control Program Commands .......................... 474
12. CDSCP Commands .................................................. 518
13. DCECP Equivalent Commands for CDSCP Commands ............... 520
14. DTSCP Commands .................................................. 616
15. DCECP Equivalent Commands for DTSCP Commands ............... 617
16. DCECP Equivalent Commands for Security Commands ............ 639
17. Locksmith Account Changes Made by the Security Server .......... 689
18. Registry Policy Changes Made by the Security Server ............ 689
About This Book

The purpose of this book is to provide complete and detailed reference information on commands that system and network administrators use when they work with the IBM z/OS Distributed Computing Environment product.

Who Should Use This Book

This book is intended for system and network administrators who understand the basic concepts of data communications and for end users of DCE and DCE applications. A knowledge of TCP/IP communications and the UNIX operating system can help you to use this guide more effectively. If you have little or no experience with Distributed Computing Environment (DCE), you should first read **z/OS DCE Introduction**.

You may find it useful to keep at hand the **z/OS DCE Administration Guide**.

How This Book Is Organized

This reference is organized into six chapters.

- **Chapter 1, “DCE and z/OS Commands” on page 1** provides information regarding the DCE and z/OS commands, **dceconf** and **modify**.
- **Chapter 2, “DCE Administration Commands” on page 11** introduces the general DCE administration commands and the DCE Control Program (**dcecp**), which provide remote access to routine DCE administrative functions.
- **Chapter 3, “Remote Procedure Call” on page 437** introduces DCE Remote Procedure Call (RPC). It also describes RPC facilities, including the RPC control program and RPC control program (**rpccp**) commands.
- **Chapter 4, “Cell Directory Service” on page 507** introduces DCE Cell Directory Service (CDS). It also provides information on CDS commands started by PROCs or CLIST and CDS control program (**cdscp**) commands. Examples of how these commands are used in local cell directories are also included in this chapter.
- **Chapter 5, “Distributed Time Service” on page 609** describes DCE Distributed Time Service (DTS). It also provides information on DTS facilities, including the DTS control program (**dtscp**) commands.
- **Chapter 6, “Security Service” on page 639** introduces DCE Security Service (SEC). It also provides information on the DCE SEC process and describes DCE Security administrative commands, including **secadmin**, **rgyedit**, and **acledit**.

**Note:** In this book the term “DCE Security Server” (or simply “Security Server”) refers to the z/OS SecureWay Security Server DCE or to a DCE Security Server provided on another host in the DCE cell. The z/OS SecureWay Security Server DCE is a component of the SecureWay Security Server for z/OS.

The chapters begin with a short introduction, and are followed by the control programs and commands, which are in alphabetic order.
Unsupported Services and Commands

The following commands and services are not supported in z/OS DCE.

- `getcellname`
- `getip`
- `cdsbrowser`
- Distributed Time Service
  - `dtsdate`
- DCE Control Program
  - `cdsalias`
  - `cellalias`
  - `cell backup`
  - `host configure`
  - `host unconfigure`
- Security Service
  - `auditd`
  - `chpss`
  - `passwd_export`
  - `passwd_import`
  - `passwd_override`
  - `sec_salvage_db`

Conventions Used in This Book

This book uses the following typographic conventions:

**Bold**

**Bold** words or characters represent system elements that you must enter into the system literally, such as commands, options, or path names.

*Italic*

*Italic* words or characters represent values for variables that you must supply.

Example font

Examples and information displayed by the system appear in constant width type style.

[]

Brackets enclose optional items in format and syntax descriptions.

{}

Braces enclose a list from which you must choose an item in format and syntax descriptions.

| A vertical bar separates items in a list of choices.

< >

Angle brackets enclose the name of a key on the keyboard.

...

Horizontal ellipsis points indicate that you can repeat the preceding item one or more times.

\

A backslash is used as a continuation character when entering commands from the shell that exceed one line (255 characters). If the command exceeds one line, use the backslash character \ as the last non-blank
character on the line to be continued, and continue the command on the next line.

This book uses the following keying conventions:

<Alt-c> The notation <Alt-c> followed by the name of a key indicates a control character sequence.

<Return> The notation <Return> refers to the key on your keyboard that is labeled with the word Return or Enter, or with a left arrow.

**Entering commands** When instructed to enter a command, type the command name and then press <Return>.

---

**Where to Find More Information**

Where necessary, this book references information in other books using shortened versions of the book title. For complete titles and order numbers of the books for all products that are part of z/OS, see the [z/OS Information Roadmap](#) SA22-7500. For complete titles and order numbers of the books for z/OS DCE, refer to the publications listed in the "Bibliography" on page 715.

For information about installing z/OS DCE components, see the [z/OS Program Directory](#).

**Softcopy Publications**

The z/OS DCE library is available on a CD-ROM, z/OS Collection, SK3T-4269. The CD-ROM online library collection is a set of unlicensed books for z/OS and related products that includes the IBM Library Reader™. This is a program that enables you to view the BookManager® files. This CD-ROM also contains the Portable Document Format (PDF) files. You can view or print these files with the Adobe Acrobat reader.

**Internet Sources**

The Softcopy z/OS publications are also available for web-browsing and for viewing or printing PDFs using the following URL:


You can also provide comments about this book and any other z/OS documentation by visiting that URL. Your feedback is important in helping to provide the most accurate and high-quality information.

**Using LookAt to Look up Message Explanations**

LookAt is an online facility that allows you to look up explanations for z/OS messages. You can also use LookAt to look up explanations of system abends.

Using LookAt to find information is faster than a conventional search because LookAt goes directly to the explanation.

LookAt can be accessed from the Internet or from a TSO command line.

You can use LookAt on the Internet at:

To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from the LookAt Web site by clicking on the News and Help link or from the z/OS Collection, SK3T-4269.

To find a message explanation from a TSO command line, simply enter: `lookat message-id` as in the following:

`lookat iec192i`

This results in direct access to the message explanation for message IEC192I.

To find a message explanation from the LookAt Web site, simply enter the message ID and select the release with which you are working.

**Note:** Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes* SA22-7624. For such messages, LookAt prompts you to choose which book to open.

**Accessing Licensed Books on the Web**

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link site:

http://www.ibm.com/servers/resourcelink

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link user ID, password, and z/OS licensed book key code. The z/OS order that you received provides a memo that includes your key code.

To obtain your IBM Resource Link user ID and password, logon to:

http://www.ibm.com/servers/resourcelink

To register for access to the z/OS licensed books:

1. Logon to Resource Link using your Resource Link user ID and password.
2. Select User Profiles located on the left-hand navigation bar.
5. Supply your key code where requested and select the Submit button.

If you supplied the correct key code you will receive confirmation that your request is being processed.

After your request is processed you will receive an e-mail confirmation.

**Note:** You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books:

1. Logon to Resource Link using your Resource Link user ID and password.
2. Select Library.
3. Select zSeries.
4. Select Software.
5. Select z/OS.
6. Access the licensed book by selecting the appropriate element.
Product Name

The product name **z/OS DCE** refers to the DCE services on z/OS.
Chapter 1. DCE and z/OS Commands

This chapter introduces you to the following commands:

- **dceconf**
  A DCE command that enables you to configure and start up DCE. This command is started from TSO/E. Refer to the [z/OS DCE Configuring and Getting Started](#) for complete details on starting and stopping `dceconf` and for the types of access privileges required.

- **modify**
  A z/OS operator command that enables you to:
  - start, stop, and query the status of DCE daemons
    - `dced`
    - `secd`
    - `cdsadv`
    - `cdsclerk`
    - `cdsd`
    - `dtstp`
    - `dtsd`
    - `auditd`
    - `pwdmgmt`
    - `gdad`
  - set the level of all debug messages
  - display or set the DCE software clock

This command can only be started from the operator console. Refer to the [z/OS DCE Administration Guide](#) for details.
dceconf

Configures and starts up DCE.

Format

dceconf

Usage

The dceconf command calls a menu-driven interface that configures and starts DCE. The menus, which consist of DCE client configuration, DCE server configurations, and deconfiguration, allow you to start individual configuration or deconfiguration routines according to the selection you enter.

This command also starts the DCE daemons.

Exit Values

In case of an error, this command repeats requests for correct input. You can exit the program from any menu.

Related Information: Books:

- z/OS DCE Administration Guide
- z/OS DCE Configuring and Getting Started
modify DCEKERN daemon

Starts, stops, or queries the status of DCE daemons.

Format

```
modify procname,command [daemon start_options] | level
```

Parameters

- **procname**: The name of the DCE kernel task. In z/OS DCE the default procname is DCEKERN.
- **command**: The action being performed on the DCE daemon or daemons. This parameter can have one of the following values:
  - **start**: Starts either a single DCE daemon or all of the DCE daemons.
  - **stop**: Stops either a single DCE daemon or all of the DCE daemons.
  - **query**: Displays the status of the DCE daemon or daemons or the release and service level of DCE.
- **daemon**: The name of the DCE daemon for which the action is being requested. This parameter can have one of the following values:
  - **dced**: The DCE host daemon.
  - **secd**: The Security Service daemon.
  - **cdsadv**: The Cell Directory Service (CDS) advertiser daemon.
  - **cdsclerk**: The CDS clerk daemon.
  - **cdsd**: The Cell Directory Server daemon.
  - **dtstp**: The time provider daemon.
  - **dtsd**: The Distributed Time Service (DTS) daemon.
  - **auditd**: The audit daemon.
  - **pwdmgmt**: The password strength daemon.
  - **gdad**: The Global Directory Agent daemon.
  - **all**: All of the DCE daemons.
- **start_options**: The values that are passed to the daemons. This parameter is valid only for the start command. The daemon configuration file contains start parameters, however if you use the following options, they are added to those options already contained in the configuration file.

  The maximum length of the start_options specified on the MODIFY command is 95 for a daemon that is to be started in a separate address space. There is no maximum length for a daemon that is to be started within the DCEKERN address space.

  This parameter can have one of the following values:
  - **dced**: There are no start_options for dced.
  - **secd**: There are no start_options for secd.
  - **cdsadv**: There are three start_options:
modify DCEKERN daemon

-c or -C cache_size
Specify the cache size in kilobytes. The default cache size is 128 kilobytes. This option is ignored by the Advertiser if it is restarted. For this option to take effect, you must stop both the CDS Clerk and Advertiser, and then start them again.

-s or -S
Specify that the CDS advertiser starts in stand-alone mode without advertising. This option is often used when operating in a wide area network (WAN) environment.

-p or -P
Receives advertisements from any cell (promiscuous mode).

cdscld
There are no start_options for cdscld.

cdscld
There are three start_options:

-a or -A
Specifies the server is the master.

-l or -L
Sets the locksmith mode. Allows the specified principal to have full access to all information stored with this server.

-v or -V
Specifies the directory version information as M.m, where M is the major directory version and m is the minor directory version. For example, enter:

-V 4.0

to use the 4.0 directory version default version.

dtstp
There are two start_options:

-p or -P period
Poll period, the number of seconds between time service queries. For example, entering 300 causes the time provider to supply the time service with the time and inaccuracy every 5 minutes.

-i or -I rate
The inaccuracy rate. This is the number of milliseconds of system inaccuracy in the timestamps delivered by the host. The default is 100 milliseconds.

These two options only apply to the DTS Null Time Provider that is shipped with DCE. If other time providers are installed, the options of those time providers will apply.

dtstd
There are two start_options:

-r or -R
Save the configuration and restore the previous configuration.

-u or -U
Do not restore the previous configuration. This overrides the -r option. It is only operative if you have previously specified -r in starting the DTS daemon.

These two options only apply to the DTS daemon that is shipped with z/OS DCE.

auditd
There are no start options for auditd.

pwdmgmt
There are no start_options for pwdmgmt.

gdad
There are seven start_options:
modify DCEKERN daemon

-\, -L  Specified to disable the LDAP connection.
Prevents LDAP from starting during gdad initialization. The
default is to start LDAP. Supported by the GDA daemon.

-b \, -B  Specifies X.500 support.
Prevents BIND from starting during gdad initialization. The
default is to start BIND. Supported by the GDA daemon.

-r \, -R  Overrides the RESOLVER_CONFIG environment variable.
Supported by the GDA daemon and refers to the BIND only.

-s \, -S  Overrides the _EUV_NAMED_CA environment variable.
Supported by the GDA daemon and refers to the BIND only.

-h \, -H  Overrides the setting of the LDAP_SERVER environment
variable. Supported by GDA daemon and ldap_addcell
utility. When there is a GDA daemon, it refers to LDAP.

-a \, -A  Overrides the setting of the LDAP_AUTH_DN environment
variable. Supported by GDA daemon and ldap_addcell
utility. When there is a GDA daemon, it refers to LDAP.

-p \, -P  Overrides the setting of the LDAP_AUTH_DN_PW
environment variable. Supported by GDA daemon and
ldap_addcell utility. When there is a GDA daemon, it refers
to LDAP.

all  There is one start_option:

-\, nodce  Starts the DCE Kernel but does not start daemons within the
DCE kernel.

level  This returns the release and service level of DCE. The level parameter is valid only for
the query command.

Usage
The modify DCEKERN daemon command manually starts or stops one or more DCE daemons, or to
view the status of the daemons. It is especially useful in situations where a daemon stops abnormally. In
z/OS DCE, the DCE daemons are contained in the DCEKERN address space, or in address spaces
created by the DCEKERN address space.

If you do start all, each daemon is started in the appropriate order without any start options, except dtsd,
which defaults to -r.

Starting and Stopping z/OS DCE Daemons: Using the modify system operator command,
you can start or stop either a single daemon or all the daemons configured on the host. However,
because the successful startup of some of the DCE daemons depends on the availability of the services
provided by other DCE daemons, there is an order in which DCE daemons must be started. The all
command starts the daemons in the correct order. See the z/OS DCE Administration Guide for more
information about the starting order of the daemons. Some daemons must run within the DCEKERN
address space, but others can be started in separate address spaces. By default, cdsd and secd are
started in separate address spaces. See the z/OS DCE Administration Guide for more information about
starting daemons in separate address spaces.
modify DCEKERN daemon

Viewing the Status of z/OS DCE Daemons: Using the `modify` system operator command, you can view the status of the DCE daemons from the operator console using the `query` option of the `modify` system command. You do not require the special privileges of a DCE administrator or an operator to use the `query` option.

Examples

- `modify DCEKERN,start all` Starts all the DCE daemons that are not currently active on the host.
- `modify DCEKERN,stop all` Stops all the DCE daemons that are currently active on the host.
- `modify DCEKERN,query all` Views the status of the DCE daemons that are currently configured on the host.

Viewing the Level of z/OS DCE DCE: Using the `modify` system operator command, you can view the release and service level information for DCE. You do not require the special privileges of a DCE administrator or an operator to use the `query` option. When you enter:

```none
modify DCEKERN,query level
```

The following displays:

```
EUVP00148I  OS/390 DCE DES Feature
Release 2.9.2 Service Level OW44972.
Created on Tue Jan 13 13:40:57 EST 2000
```

Related Information: Books:

- [z/OS DCE Administration Guide](#)
modify DCEKERN debug

Dynamically sets the level of all debug messages.

Format

modify procname.debug
action=\{on | off | mod | print\}
\{entity=\{dcectrl | cdssd | cdssadv | cdsclerk | dtsp | secd | dced | dtstp | dtsd | auditd | pwdmgmt\} | pid=process_ID\}
\{comp=\{component_name | *\}\}
\{subcomp=\{sub-component_name | *\}\}
\{level=\{0-9\}\}
\{mask=\{0x000-0x3FF\}\}

Options

[a]ction=  The action to take. This parameter can have one of the following values:
  on        Activates the debug messages.
  off       De-activates the debug messages.
  mod       Modifies the current debug level.
  print     Prints the contents of the debug message cache to output destination.

Note:  An [e]ntity or [p]id must be specified for every action.

[e]ntity=  The name of the DCE process on which to perform the action. This parameter can have
           one of the following values:
         dcectrl        The DCE control task.
         secd           The Security Service daemon.
         dced           The DCE daemon.
         cdssd          The CDS daemon.
         cdssadv        The CDS advertiser daemon.
         cdsclerk       The CDS clerk daemon.
         dtstp          The DTS time provider daemon.
         dtspd          The DTS daemon.
         auditd         The audit daemon.
         gdad           The GDA daemon.
         pwdmgmt       The password strength daemon.

[p]id=       The process_ID of the user application.

[c]omp=       The name of the component to be modified. This parameter is valid only for the mod
              action, and can have one of the following values:
               component_name    The name of the individual component.
               *                  All components within the entity.

[s]ubcomp=   The name of the sub-component to be modified. This parameter is valid only for the mod
             action, and can have one of the following values:
modify DCEKERN debug

sub-component_name  The name of the individual sub-component.
*  All sub-components within the entity.

[level=]  The level to which the debug level is set. This parameter is valid only for the mod action, and can have one of the following values:
0  Debug level is set to off.
1-9  Debug level is set to the level number specified.

[mask=]  The hexadecimal constant with a binary representation used to selectively enable or disable debug levels that are less than or equal to the current debug level setting. Each bit, numbered right-to-left, 0 through 9, corresponds to a debug level 0 through 9. Output from a specific debug level is enabled if its corresponding bit position is set (1) or disabled (0). The mask will not alter the action of debug level 0 (off). The default value is 0x3FF. This option is valid only for the mod action.

Parameters

procname  The name of the DCE kernel task. In z/OS DCE the default procname is DCEKERN.

Usage

The modify DCEKERN debug command activates or de-activates debugging, and to dynamically set the level of all debug messages. After the environment variables and routing files are checked to establish the initial debug levels, the levels indicated by the mod parameter are set. The debug levels that are set remain in effect as long as the process is running, even if debugging is turned off and on.

Examples

1. The following command activates debugging for the DTS daemon:
   modify DCEKERN,debug action=on entity=dtsd

2. The following command activates debugging for a user application running in an address space with the process ID of 14521:
   modify DCEKERN,debug action=on pid=14521

3. The following commands set the mutex sub-component to debug level 9 for the DTS daemon:
   modify DCEKERN,debug action=mod entity=dtsd comp=thr subcomp=mutex level=9
   or
   modify DCEKERN,debug a=mod e=dtsd c=thr s=mutex l=9

4. The following command sets the debug level to 3 for all components of the DTS daemon:
   modify DCEKERN,debug action=mod entity=dtsd comp=** subcomp=** level=3

5. The following command prints the contents of the debug message cache, if enabled, for the DTS daemon:
   modify DCEKERN,debug a=print e=dtsd

6. The following command sets the debug level to 9 for all components of the DTS daemon, and disables output from all levels except 9 and 1:
   modify DCEKERN,debug action=mod entity=dtsd comp=** subcomp=** level=9 mask=0x201

Related Information:  Books:

- z/OS DCE Administration Guide
modify DCEKERN clock

Displays or sets the DCE software clock relative to Greenwich mean time (GST).

Format

modify proctype,clock command

Parameters

proctype The name of the DCE kernel task. In z/OS DCE the default proctype is DCEKERN.
command The action being performed on the clock. This parameter can have one of the following values:
    display Displays the DCE software clock, shows adjustment, time zone, and clock offset information.
    reset Resets the DCE software clock to exactly the same value as that of the z/OS hardware clock.
    set time Sets the time to the specified time, where time is an absolute time in ISO-compliant format.
    setrel relative-time Adds the specified relative time to the DCE software clock.
    setdst N Sets the daylight savings time of the clock, where N is an integer to which the daylight savings time is set.

Usage

The modify DCEKERN clock command displays clock details, resets the software time back to the hardware time, and selects Daylight Savings Time information.

The z/OS DCE software clock consists of the machine hardware time and a logical offset from the hardware time. This software clock is manipulated by the DCE distributed time service (DTS). Issuing the reset command affects the operation of the DTS component.

Examples

- To display the DCE clock:
  modify DCEKERN,clock display
- To set the daylight savings field of the DCE clock to 1:
  modify DCEKERN,clock setdst 1
- To set the clock to a specified absolute time:
  modify DCEKERN, clock set 1994-12-16-10:25:23.000-04:00
- To add a relative time to the DCE software clock:
  modify DCEKERN, clock setrel 04:30

Related Information: Books:

- [z/OS DCE Administration Guide](#)
modify DCEKERN clock
Chapter 2. DCE Administration Commands

This chapter describes the DCE administration commands that are general to DCE rather than specific to a particular component:

**dcecp** The DCE control program is the primary DCE administration interface, providing remote access to routine DCE administrative functions.

The **dcecp** program supports the following **dcecp** administration objects. For complete descriptions of the administration objects, refer to the individual commands.

<table>
<thead>
<tr>
<th>Administration Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>account</td>
<td>Manages an account in the DCE Security Service registry.</td>
</tr>
<tr>
<td>acl</td>
<td>Creates, modifies, and removes access control lists.</td>
</tr>
<tr>
<td>attrlist</td>
<td>Manipulates attribute lists.</td>
</tr>
<tr>
<td>aud</td>
<td>Manages the audit daemon (<strong>audid</strong>) on a DCE host.</td>
</tr>
<tr>
<td>audevents</td>
<td>Manages audit event classification on a DCE host.</td>
</tr>
<tr>
<td>audfilter</td>
<td>Manages audit event filters on a DCE host.</td>
</tr>
<tr>
<td>audtrail</td>
<td>Controls the output format for audit events captured on a DCE host.</td>
</tr>
<tr>
<td>cdscache</td>
<td>Manages the CDS clerk cache on a DCE host.</td>
</tr>
<tr>
<td>cell</td>
<td>Manages cell configuration information.</td>
</tr>
<tr>
<td>clearinghouse</td>
<td>Manages CDS clearinghouse operations in a DCE cell.</td>
</tr>
<tr>
<td>clock</td>
<td>Manages the clock on a local DCE host.</td>
</tr>
<tr>
<td>directory</td>
<td>Manages directories in the DCE Cell Directory Service.</td>
</tr>
<tr>
<td>dts</td>
<td>Manages Distributed Time Service servers and clerks.</td>
</tr>
<tr>
<td>endpoint</td>
<td>Manages endpoint information in a DCE host's endpoint map workload balancing for Parallel Sysplex® environments.</td>
</tr>
<tr>
<td>group</td>
<td>Manages groups in the DCE Security Service registry.</td>
</tr>
<tr>
<td>host</td>
<td>Manages hosts within a cell.</td>
</tr>
<tr>
<td>hostdata</td>
<td>Manages a DCE host's principal name and cell affiliation information on the host.</td>
</tr>
<tr>
<td>keytab</td>
<td>Manages server passwords on DCE hosts.</td>
</tr>
<tr>
<td>link</td>
<td>Manages softlinks in the DCE Cell Directory Service.</td>
</tr>
<tr>
<td>log</td>
<td>Manages routing for DCE serviceability messages.</td>
</tr>
<tr>
<td>name</td>
<td>Manipulates names in the DCE name space.</td>
</tr>
<tr>
<td>object</td>
<td>Manages object entries in the DCE Cell Directory Service.</td>
</tr>
<tr>
<td>organization</td>
<td>Manages organizations in the DCE Security Service registry.</td>
</tr>
<tr>
<td>principal</td>
<td>Manages principals in the DCE Security Service registry.</td>
</tr>
<tr>
<td>registry</td>
<td>Manages the database of a DCE Security Service registry.</td>
</tr>
<tr>
<td>rpcentry</td>
<td>Manages a server entry stored in the DCE Cell Directory Service.</td>
</tr>
<tr>
<td>rpcgroup</td>
<td>Manages a group entry stored in the DCE Cell Directory Service.</td>
</tr>
<tr>
<td>rpcprofile</td>
<td>Manages a profile entry stored in the DCE Cell Directory Service.</td>
</tr>
<tr>
<td>secval</td>
<td>Manages the security validation service in <strong>dced</strong>.</td>
</tr>
<tr>
<td>server</td>
<td>Manages servers and their configuration information on DCE hosts.</td>
</tr>
</tbody>
</table>
The following DCE control program commands are not supported in z/OS DCE:

- cdsalias
- cellalias

**Related Information:** See each command for further information. The `dcecp` command and each object acted on by `dcecp` are listed alphabetically in this section. General information about `dcecp` is included in the `dcecp` section.

<table>
<thead>
<tr>
<th>Table 1 (Page 2 of 2). DCECP Administration Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>user</strong></td>
</tr>
<tr>
<td><strong>utc</strong></td>
</tr>
<tr>
<td><strong>uuid</strong></td>
</tr>
<tr>
<td><strong>xattrschema</strong></td>
</tr>
</tbody>
</table>
account

A dcecp object that manages an account in the DCE Security Service.

Format

account catalog [server_name_list] [-simplename]

account create account_name_list [-attribute attribute_list] [-password password] [-group group_name] [-organization organization_name] [-mypwd password] [other_attribute_options]

account delete account_name_list

account generate account_name

account help [operation] [-verbose] [-syntax]

account modify account_name_list {-change attribute_list | attribute options [-mypwd password]}

account operations

account show account_name_list [-policies] [-all]

Parameters

account_name The name of an account to act on. It must be an account name, not the name of the registry database object that contain registry information about the account. The name can be a fully qualified account name as in:

/.../cellname/account_name

Or, it can be a cell-relative account name as in:

account_name

A cell-relative name refers to an account in the cell named in the _s(sec) convenience variable or in the default cell of the local host, if the _s(sec) convenience variable is not set.

account_name_list A list of one or more names of accounts to act on.

Do not mix full path names with simple names in a list, and do not use names of database objects that contain registry information about accounts (account names must not begin with /./sec/account/).

operation The name of one specific account operation (subcommand) about which you want help information.

server_name_list A list of one or more names of registry servers to act on. The names can be fully qualified as in:

/.../cellname/subsys/dce/sec/registry_name

Or, they can be cell-relative names as in:

subsys/dce/sec/registry_name
account

Usage

The **account** object represents registry accounts. Each account is associated with one principal, one group and one organization. However, accounts are named by just the principal name. Aliases are differentiated for principals, so one principal can have multiple accounts under different alias names. The account has attributes to specify the group and organization of the account, these must be specified on account creation.

After this command runs, the `_b(sec)` convenience variable is set to the name of the server that was bound to for the command. The value of the variable `_s(sec)` before the command is treated as a hint; the server specified is contacted if it can service the request. A case where it cannot service the request is if a read-only registry was bound to, and the next command is a create, modify, or delete command. In this case, the master registry is bound to automatically and the `_b(sec)` variable updated appropriately. The value of the variable is the name of the registry bound to in one of the formats specified as valid for the argument to the **registry** object.

Attributes

The **account** object supports two kinds of attributes, account and policy.

- Account attributes may or may not have default values. They assume a default value or a value set by administrators.
- Policy attributes regulate such things as account and password lifetimes. Policy attributes have registry-wide default values. Policy attributes always assume the most restrictive value whether it is the registry-wide default value or a value set by administrators.

The account attributes are:

- **acctvalid {yes | no}**
  A flag set to determine account validity. Values are either yes or no. An account with an acctvalid attribute set to no is not valid and cannot be logged into. The default is yes.

- **client {yes | no}**
  A flag set to specify whether the account is for a principal that can act as a client. Values are either yes or no. If you set this flag to yes, the principal is able to log into the account and acquire tickets for authentication. The default is yes.

- **created creators_name ISO_timestamp**
  A list of two items. The first is the principal name of the creator of the account. The second is an ISO timestamp showing the time of creation. This attribute is set by the system and may not be modified.

- **description**
  A text string (limited to Portable Character Set (PCS)) that describes the use of the account.

- **dupkey {yes | no}**
  A flag set to determine if tickets issued to the account principal can have duplicate keys. Values are either yes or no. The default is no.

- **expdate date**
  The date on which the account expires. Change the date in this field to renew the account. Specify the time using an ISO compliant time format such as CCYY-MM-DD-hh:mm:ss or the string none. The default is none.

- **forwardabletkt {yes | no}**
  A flag set to determine whether a new ticket-granting ticket with a network address that differs from the present ticket-granting ticket network address can be issued to the
account principal. The **proxiabletkt** attribute performs the same function for service tickets. Values are either **yes** or **no**. The default is **yes**.

**goodsince date time**

The date and time the account was last in an uncompromised state. Any tickets granted before this date are not valid. The value is an ISO timestamp. When the account is initially created, the **goodsince** date is set to the current date. Control over this date is useful if you know that an account password was compromised. Changing the password can prevent the unauthorized principal from accessing the system again using that password, but it does not prevent the principal from accessing the system components for which tickets were obtained fraudulently before the password was changed. To eliminate the principal's access to the system, the tickets must be canceled.

**group group_name**

The name of the group associated with the account. The value is a single group name of an existing group in the registry. This attribute must be specified on an **account create** command. It does not have a default value. If this group is deleted from the registry, then the account is also deleted.

**home directory_name**

The file system directory in which the principal is placed in at login.

**lastchange principal_name ISO_timestamp**

A list of two items. The first is the principal name of the last modifier of the account; the second is an ISO timestamp showing the time of the last modification. This attribute is set by the system and may not be modified directly.

**organization organization_name**

The name of the organization associated with the account. The value is a single organization name of an existing organization in the registry. This attribute must be specified on an **account create** command. It does not have a default value. If this organization is deleted from the registry, then the account is also deleted.

**password password**

The value of this attribute is the password of the account. There is no default value, so this attribute must be specified in an **account create** command. This attribute is not returned by an **account show** command.

When you change your DCE password, if you are enrolled in z/OS DCE single sign-on support, you must use the **storepw** command to change your DCE password in the RACF® database. (RACF is a component of the SecureWay Security Server for z/OS.) Failing to do so causes single sign-on to fail. See the **storepw** command for more information.

**postdatedtkt {yes | no}**

A flag set to determine if tickets with a start time specified in the future can be issued to the account principal. Values are either **yes** or **no**. The default is **no**.

**proxiabletkt {yes | no}**

A flag set to determine whether a new ticket with a different network address than the present ticket can be issued to the account's principal. The **forwardabletkt** attribute performs the same function for ticket-granting tickets. Values are either **yes** or **no**. The default is **no**.

**pwdvalid {yes | no}**

A flag set to determine whether the current password is valid. If this flag is set to **no**, the system prompts the principal to change the password the next time a principal logs
account

into the account. (Note that this flag is separate from the pwdexpdate policy, which
sets time limits on password validity.) Values are either yes or no. The default is yes.

renewabletkt {yes | no}
A flag set to determine if the ticket-granting ticket issued to the account principal can be
renewed. If this flag is set to yes, the Authentication service renews the ticket-granting
ticket if its lifetime is valid. Values are either yes or no. The default is yes.

server {yes | no}
A flag set to specify whether the account is for a principal that can act as a server. If
the account is for a server that engages in authenticated communications, set this flag
to yes. Values are either yes or no. The default is yes.

shell path_to_shell
The path of the shell that is run when a principal logs in.

stdtgtauth {yes | no}
A flag set to determine whether service tickets issued to the account principal use the
standard DCE ticket-granting ticket authentication mechanism. Values are either yes or
no. The default is yes.

The policy attributes are:

maxtktlife relative_time
The maximum amount of time that a ticket can be valid. Specify the time using the
DTS relative time format ([-]DD-hh:mm:ss). When a client requests a ticket to a server,
the lifetime granted to the ticket takes into account the maxtktlife set for both the
server and the client. In other words, the lifetime cannot exceed the shorter of the
server or client maxtktlife. If you do not specify a maxtktlife for an account, the
maxtktlife defined as registry authorization policy is used.

maxtktrenew relative_time
The amount of time before a principal ticket-granting ticket expires and that principal
must log in again to the system to re-authenticate and obtain another ticket-granting
ticket. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss). The
lifetime of the principal service tickets can never exceed the lifetime of the principal's
ticket-granting ticket. The shorter you make this, the greater the security of the system.
However, because principals must log in again to renew their ticket-granting ticket, the
time needs to consider user convenience and the level of security required. If you do
not specify this for an account, the maxtktrenew lifetime defined as registry
authorization policy is used.

See the [z/OS DCE Administration Guide] for more information about attributes.

Related Information

Commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>organization</td>
</tr>
<tr>
<td>group</td>
<td>registry</td>
</tr>
<tr>
<td>principal</td>
<td>storepw</td>
</tr>
</tbody>
</table>
account catalog

Returns a list of the names of all accounts in the registry.

Format

account catalog [server_name_list] [-simplename]

Options

-simplename Returns a list of account names in the registry that are relative, not fully qualified.

Usage

The account catalog operation returns a list of the names of all accounts in the registry database. This command takes an optional server name argument to specify the server to use. By default, fully qualified names are returned in the form cellname/accountname. If the -simplename option is given, then the account name is relative, not fully qualified.

Privilege Required: You must have r (read) permission to the principal named in the account.

Examples

dcecp> account catalog -simplename
nobody
root
daemon
uucp
bin
dce-ptgt
dce-rgy
krbtgt/goodco.com

cell_admin
hosts/pmin17/self
hosts/pmin17/cds-server
hosts/pmin17/gdaward

dcecp>

dcecp> account catalog
/.../mycell.goodco.com/nobody
/.../mycell.goodco.com/root
/.../mycell.goodco.com/daemon
/.../mycell.goodco.com/uucp
/.../mycell.goodco.com/bin
/.../mycell.goodco.com/dce-ptgt
/.../mycell.goodco.com/dce-rgy
/.../mycell.goodco.com/krbtgt/mycell.goodco.com
/.../mycell.goodco.com/cell_admin
/.../mycell.goodco.com/hosts/pmin17/self
/.../mycell.goodco.com/hosts/pmin17/cds-server
/.../mycell.goodco.com/hosts/pmin17/gdaward
/.../mycell.goodco.com/ward

dcecp>
account catalog

Related Information

Commands:

dcecp  principal  registry
group  organization  rgyedit
account create

Creates a new account in the registry database.

Format

    account create account_name_list [-attribute attribute_list | -password password -group group_name -organization organization_name [-mypwd password] [other_attribute_options]]

Options

-attribute attribute_list
    Lets you specify attributes using an attribute list rather than using the individual attribute options such as -group, -organization, and so on. The format is:

    \{\{attr attr_value\} \{attr attr_value\} . . . \{attr attr_value\}\}

    You must specify the group, organization, and password attributes on the command line. See "Attributes" on page 14 for a description of the group, organization, password, and other attributes.

-attribute_option attribute_value
    As an alternative to using the -attribute option with an attribute list, you can create individual attribute options by adding a hyphen (-) to any attributes listed in "Attributes" on page 14. The -attribute option is intended to use in scripts when you can paste in lengthy attribute lists from previous commands. The individual attribute options may be easier to use for interactive commands.

    If you specify attributes using individual attribute options, you must use at least the group, organization, and password options. See "Attributes" on page 14 for a description of the group, organization, and password attributes.

-group group_name
    You must associate a group with the account. You can use the -group option or you can create it using the group attribute with the -attribute option and an attribute_list.

-mypwd password
    Lets you enter your password. You must enter your password to create an account. This check prevents an existing privileged session to create unauthorized accounts. You can use the -mypwd option. You can specify it by using the mypwd attribute with the -attribute option and an attribute_list. If you do neither, you are prompted for your password.

-organization organization_name
    You must associate an organization with the account. You can use the -organization option or you can create it using the organization attribute with the -attribute option and an attribute_list.

-password password
    You must create a password for the account. You can use the -password option or you can create it using the password attribute with the -attribute option and an attribute_list.
account create

Usage

The **account create** operation creates a new account. The argument is a list of names of accounts being created. An empty string is returned when successful. Use options to specify the attributes of the newly created account. You must specify the **group**, **organization**, and **password** attributes on the command line (either in an attribute list or with attribute options). You are prompted for your password if you do not specify the **mypwd** option. You must be logged into the cell in which the new account is being created. Supports both attribute lists (with an **-attribute** option) and attribute options named after each attribute and policy. All options are applied to all accounts in the argument.

To protect the account password being entered, enter the **account create** command only from within the **dcecp** program. You cannot enter this command from the operating system prompt using **dcecp** with the **-c** option.

Before you can create a new account, you must create a principal using the **principal create** command. Then you must add the principal to an existing group and organization using the **group add** and **organization add** commands.

**Privilege Required:** You must have **gmau** (**groups**, **mgmt_info**, **auth_info**, **user_info**) permissions to the principal named in the account, **rtM** (**read**, **test**, **Member_list**) permissions to the organization named in the account, **tM** (**test**, **Member_list**) permissions to the group named in the account, and **r** (**read**) permission on the registry policy object.

**Examples**

```bash
dcecp> principal create John_Hunter
dcecp> group add users -member John_Hunter
dcecp> organization add users -member John_Hunter
dcecp> account create John_Hunter -group users -organization users \
> -mypwd my.secret.password -password change.me
dcecp>
```

**Related Information**

Commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td></td>
</tr>
<tr>
<td>group</td>
<td>principal</td>
</tr>
<tr>
<td></td>
<td>organization</td>
</tr>
<tr>
<td></td>
<td>registry</td>
</tr>
<tr>
<td></td>
<td>registryedit</td>
</tr>
</tbody>
</table>
account delete

Deletes accounts from the registry.

**Format**

```
account delete account_name_list
```

**Usage**

The `delete` operation deletes accounts from the registry. The argument is a list of names of accounts to be deleted. If the accounts do not exist, an error is generated. An empty string is returned when successful.

**Privilege Required:** You must have `rmau (read, mgmt_info, auth_info, user_info)` permissions for the principal named in the account.

**Examples**

```
dcecp> account delete John_Hunter
```

**Related Information**

Commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>principal</td>
<td>registry</td>
</tr>
<tr>
<td>group</td>
<td>organization</td>
<td>rgyedit</td>
</tr>
</tbody>
</table>
account generate

Generates a random password for an account.

Format
account generate account_name

Usage
The account generate operation generates a random password for an account. The argument is the name of the account for which the password is generated.

To run the account generate command, the pwd_strength server must be running, the principal specified in the command must exist, and the pwd_mgmt_binding and pwd_val_type Extended Registry Attributes must be attached to the principal, otherwise, an error is generated.

After the password is generated, run the account create command to establish the account, or the account modify command to change the account password, supplying the randomly generated password as the account's password.

Privilege Required: You must have gmau (groups, mgmt_info, auth_info, user_info) permissions for the principal named in the account.

Examples

dcecp> account generate John_Hunter
x23cq4ti

dcecp>

Related Information

Commands:
dcecp principal registry
group organization rgyedit
account help

Returns help information about the account object and its operations.

Format

account help [operation | -verbose] [-syntax]

Options

-syntax          Displays the syntax diagram of each account operation.
-verbose         Displays information about the account object.

Usage

The account help command is used without an argument or option to return brief information about each account operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the account object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required:    No special privileges are needed to use the account help command.

Examples

dcecp> account help
    catalog          Returns the names of all accounts in the registry.
    create           Creates an account in the registry.
    delete           Deletes an account from the registry.
    generate         Generates a random password for an account in the registry.
    modify           Modifies an account in the registry.
    show             Returns the attributes of an account.
    help             Prints a summary of command-line options.
    operations       Returns the valid operations for command.

Related Information

Commands:

  dcecp    principal
  group    organization
  registry
  registry
  rgoryedit

Chapter 2. DCE Administration Commands  23
account modify

Changes attributes and policies of accounts.

Format

```
account modify account_name_list
{-change attribute_list | attribute options [-mypwd password]}
```

Options

-attribute_option attribute_value

As an alternative to using the -attribute option with an attribute list, you can create individual attribute options by adding a hyphen ( - ) to any attributes listed in “Attributes” on page 14 The -attribute option is intended for use in scripts when you can paste in lengthy attribute lists from previous commands. The individual attribute options may be easier to use for interactive commands.

-change attribute_list

Lets you specify attributes using an attribute list rather than using the individual attribute options such as -group, -organization, and so on. The format is:

```
{attr attr_value} {attr attr_value} . . . {attr attr_value}
```

-mypwd password

Lets you supply your password when changing the account password. -mypwd is required only when changing the account password. This check prevents using an existing privileged session to modify accounts. You cannot specify the -mypwd option with the -change option. Instead, specify the mypwd attribute in the -change attribute_list. If the -mypwd option or attribute is not specified, you are prompted for your password.

Usage

The account modify operation changes information about an account. The -add and -remove options are not supported, as each attribute associated with an account must always have a value. However, the -change option can be used to modify the value of any of the modifiable attributes. The value of the -change option is an attribute list describing the new values for the specified attributes.

The -change option can be used to modify the value of any attribute using an attribute list format. You can also use attribute options of the form -forwardabletkt {yes | no}.

To protect any passwords being entered, the account modify command can be entered only from within the dcecp program. You cannot enter this command from the operating system prompt using dcecp with the -c option, if you are changing the account password.

Privilege Required: You must have rm (read, mgmt_info) permissions for the principal named in the account.

Examples
account modify

dcecp> account modify John_Hunter -expdate 1997

dcecp> account show John_Hunter

{acctvalid yes}
{client yes}
{created /.../mycell.goodco.com/cell_admin1996-06-15-18:31:08.000+00:00I-----}
{description{}}
{dupkey no}
{expdate 1997-06-16-00:00:00.000+00:00 I-----}
{forwardabletkt yes}
{goodsince 1996-06-15-18:31:05.000+00:00I-----}
{group users}
{home /}
{lastchange /.../mycell.goodco.com/cell_admin1996-06-16-12:21:07.000+00:00I-----}
{organization users}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{rene wabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}

dcecp>

Related Information

Commands:

dcecp             principal             registry
group             organization         rgyedit

Chapter 2. DCE Administration Commands
account operations

Returns a list of the operations supported by the account object.

Format
account operations

Usage

The account operations command uses no arguments, and returns a list of the available operations for the account object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the account operations command.

Examples

dcecp> account operations
catalog create delete generate modify show help operations
dcecp>

Related Information

Commands:

dcecp                principal                registry
  group               organization             rgymedit
account show

Shows attribute information for the specified accounts.

Format

account show account_name_list [-policies] [-all]

Options

-all Return the attributes followed by the policies.
-policies Returns only the policies of the account, not other attributes.

Usage

The account show operation returns an attribute list describing the specified accounts. The argument is a list of names of accounts to be operated on. If more than one account is given, the attributes and policies are concatenated with a blank line between accounts. The -policies option lets you return the policies of the account instead of the attributes. The -all option returns the attributes followed by the policies.

Attributes and policies are returned in alphabetic order. If the account does not have any policies, then the string nopolicies is returned. If the cell and principal names cannot be obtained for the created or lastchange attributes, then the UUIDs of these display, instead.

The policies that are actually in effect can be different from the account policies if the registry-wide policies are more restrictive. If this is the case, the show command alters the attribute structure on output to include an effective tag and the effective value, much in the same way organization show does.

Privilege Required: You must have r (read) permission to the principal named in the account.

Examples
account show

dcecp> account show John_Hunter
{acctvalid yes}
{client yes}
{created /.../mycell.goodco.com/cell_admin
1996-06-15-18:31:08.080+00:00I-----}
{description {}}
{dupkey no}
{expdate 1997-06-16-00:00:00.000+00:00I-----}
{forwardabletkt yes}
{goodsince
1996-06-15-18:31:05.000+00:00I-----}
{group users}
{home /}
{lastchange /.../mycell.goodco.com/cell_admin
1996-06-16-12:21:07.000+00:00I-----}
{organization users}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}
dcecp>

Related Information

Commands:
dcecp group principal registry
group organization rgyedit
A dcecp object that manages DCE access control lists.

**Format**

- `acl check acl_name_list [ -entry ] [ -type acl_manager_name ]`
- `acl delete acl_name_list [ -ic | -io | -entry ] [ -type acl_manager_name ]`
- `acl help [ operation ] [ -verbose ] [ -syntax ]`
- `acl modify acl_name_list [ -ic | -io | -entry ] [ -type acl_manager_name ]`  
  `[-cell new_cell_name ] [-purge ]`
  `-add acl_entry_list_with_permissions [ -mask { calc | nocalc } ]`
  `-change acl_entry_list_with_permissions [ -mask { calc | nocalc } ]`
  `-remove acl_entry_list_without_permissions ]`
- `acl operations`  
- `acl permissions acl_name_list [ -ic | -io | -entry ] [ -type acl_manager_name ]`
- `acl replace acl_name_list -acl acl_entry_list [ -ic | -io | -entry ] [ -type acl_manager_name ]`  
  `[-cell new_default_cellname ]`
- `acl show acl_name_list [ -ic | -io | -entry ] [ [ -type acl_manager_name ] [ -cell ] ] [ -managers ]`

**Parameters**

- `acl_name_list` A list of one or more names of ACLs to act on. Use `-io`, `-ic`, and `-entry` options to distinguish the correct ACL. An ACL name can be a regular object name, for example: `/./sales/printers/text_printers`

  You can also use a list of string bindings with residual names appended to identify the objects. The residual name specifies whether the object is a principal, group, or organization by supplying its principal, group, or organization name.

  There are four possible formats you can use to specify a string binding.

  In string syntax, you can use:

  `{uuid@prot_seq:net_addr residual_name}`

  For example,

  `acl check {006f859c-ed3d-1d57-a383-0000c0239a70@ncacn_ip_tcp:130.105.5.45 principal/aaa}`

  or

  `{uuid@prot_seq:net_addr[endpoint] residual_name}`

  In TCL syntax, you can use:

  `{uuid prot_seq net_addr residual_name}`

  For example,

  `acl check {006f859c-ed3d-1d57-a383-0000c0239a70 ncacn_ip_tcp 130.105.5.45 principal/aaa}`

  or

  `{uuid prot_seq net_addr endpoint residual_name}`

- `operation` The name of one specific acl operation (subcommand) about which you want help information.
Data Structures
An ACL entry has the following syntax:

type key permissions

where:

type Identifies the role of the ACL entry.
key Identifies the specific object to which the entry applies. For an entry type of extended, key contains the ACL data.
permissions The ACL permissions.

The dcecp syntax of ACL entries is a list of two or three elements. The first element is the type, the optional second element is the key, and the last element is the set of permission bits. The permission bits are represented as a single character if the permission is granted and as a hyphen (-) if it is not. An ACL is a list of ACL entries. An example of an ACL is as follows:

{unauthenticated -r-----}
{user_obj crwx---}
{user britten crwx---}
{user mahler -rwx---}
{foreign_user /.../C=US/O=OSF/OU=dce/pro/bach crwxidt}
{group_obj -rwx---}
{group dds -rwx---}
{any_other -r-----},
{extended c417faf8-83470065-c864-ace3-c3e008c70001e5559bb.a.b.c.a1.4.0a0b0c0d -rwx---}

On output, the above syntax is used with one addition. If masking produces ineffective bits in an ACL entry, the entry has two additional elements. The first is the string effective and the second element is the set of effective permissions. This is added only for those ACL entries that have ineffective bits. For example:

{mask_obj -r-----}
{user_obj crwx---}
{user britten crwx--- effective -r-----}

On input, do not include the string effective or the effective permissions. You can enter permissions in any order, omitting hyphens (-) for permissions not granted. For example, if you want to enter the above ACL to a command, enter:

{mask_obj r}
{user_obj crwx}
{user britten wcwx}

Defined ACL Entry Types

any_other Permissions for all authenticated principals unless those principals match a more specific entry in the ACL.
any_other_delegate Delegated permissions for all authenticated principals unless those principals match a more specific entry in the ACL.
extended A special entry that allows client applications running at earlier DCE versions to copy ACLs to and from ACL managers running at the current DCE version without losing any data. The extended entry allows the application running at the earlier version to obtain a printable form of the ACL. The extended ACL entry has the following form:
extended `uuid.ndr.ndr.ndr.ndr.number_of_byte.data`

where:

`uuid` Identifies the type extended ACL entry. This UUID can identify one of the ACL entry types described here, or an as-yet-undefined ACL entry type.

`ndr.ndr.ndr.ndr` Up to four Network Data Representation (NDR) format labels (in hexadecimal format and separated by periods) that identify the encoding of data.

`number_of_bytes` A decimal number that specifies the total number of bytes in `data`.

`data` The ACL data in hexadecimal form. Each byte of ACL data is two hexadecimal digits. The ACL data includes all of the ACL entry specifications except the permissions (described later) that are entered separately. The data is not interpreted; it is assumed that the ACL Manager to which the data is being passed can understand that data.

`foreign_group` Permissions for a specific group in a foreign cell. This type of ACL entry must include a key that identifies the specific group and the group cell.

`foreign_group_delegate` Delegated permissions for a specific, authenticated group in a foreign cell. This type of ACL entry must include a key that identifies the specific group and the group's cell.

`foreign_other` Permissions for all authenticated principals in a specific foreign cell, unless those principals are specifically named in an ACL entry of type `foreign_user` or are members in a group named in an entry of type `foreign_group`. This type of ACL entry must include a key that identifies the specific foreign cell.

`foreign_other_delegate` Delegated permissions for all authenticated principals in a specific foreign cell, unless those principals are specifically named in an ACL entry of type `foreign_user`, or `foreign_user_delegate`, or members in a group named in an entry of type `foreign_group`, or `foreign_group_delegate`. This type of ACL entry must include a key that identifies the specific foreign cell.

`foreign_user` Permissions for a specific, authenticated user in a foreign cell. This type of ACL entry must include a key that identifies the specific principal and the principal's cell.

`foreign_user_delegate` Delegated permissions for a specific, authenticated user in a foreign cell. This type of ACL entry must include a key that identifies the specific principal and the principal's cell.

`group` Permissions for a specific group in the ACL’s default cell. This type of ACL entry must include a key that identifies the specific group.

`group_delegate` Delegated permissions for a specific group in the ACL cell. This type of ACL entry must include a key that identifies the specific group.

`group_obj` Permissions for the object’s real or effective owning group.

`group_obj_delegate` Delegated permissions for the object's real or effective group.

`mask_obj` Permissions for the object mask that is applied to all entry types except `user_obj`, `other_obj`, and `unauthenticated`.

`other_obj` Permissions for others in the local cell who are authenticated but are not named by a more specific entry type.
other_obj_delegate
Delegated permissions for others in the local cell who are not named by a more specific entry type.

unauthenticated
Maximum permissions applied when the accessor does not pass authentication procedures. This entry is used for principals that failed authentication because of bad keys, principals who are entirely outside of any authentication cell, and principals who choose not to use authenticated access. Permissions granted to an unauthenticated principal are masked with this entry, if it exists. If this entry does not exist, access to unauthenticated principals is always denied.

user
Permissions for a specific authenticated principal in the ACL's default cell. This type of ACL entry must include a key that identifies the specific principal.

user_delegate
Delegated permissions for a specific principal user in the ACL cell. This type of ACL entry must include a key that identifies the specific principal.

user_obj
Permissions for the object's real or effective owner.

user_obj_delegate
Delegated permissions for the object's real or effective user.

Key
The key identifier (principal or group name) specifies the principal or group to which the ACL entry applies. For entries of entry type extended, key is the data passed from one ACL Manager to another. A key is required for the following types of ACL entries:

foreign_user
Requires a fully qualified cell name in addition to the principal name.

foreign_group
Requires a fully qualified cell name in addition to the group name.

foreign_other
Requires a fully qualified cell name.

foreign_user_delegate
Requires a fully qualified cell name, the principal name, and a key that identifies the principal and the principal's cell.

foreign_group_delegate
Requires a fully qualified cell name, the group name, and a key that identifies the group and the group's cell.

group
Requires a group name only.

user
Requires a principal name only.

Permissions
The permissions argument specifies the set of permissions that defines the access rights conferred by the entry. Because each ACL manager defines the permission tokens and meanings appropriate for the objects it controls, the actual tokens and their meanings vary. For example, the Distributed File Service, the Directory Service, and the Security Registry Service each implement a separate ACL manager, and each can use a different set of tokens and permissions. This means that file system objects, objects in the name space, and registry objects can each use different permissions. Use the acl permissions operation to display the currently available tokens and their meanings. See the documentation for the DCE component you are using to obtain a more detailed description of its specific permissions.
Usage

The `acl` object represents an Access Control List that may exist on any object such as a server, interface, name service entry, container (directory), or file.

ACLs consist of ACL entries. ACL entries are visible only as members of ACLs. There is no object that represents ACL entries, only the `acl` object representing an entire ACL. Most of the `acl` operations deal directly with the ACL. See "Data Structures" on page 30 for a description of the syntax of ACLs and ACL entries. An ACL has one attribute called `cell` that represents the default cell of the ACL.

In most cases, the name of an object also specifies the name of the associated ACL to manipulate. However, some objects have more than one ACL and some names can refer to more than one object. These vague differences are resolved by using various options on the command line.

An object can have more than one ACL. For example, container objects (such as CDS directories and directories in the registry) have three ACLs:

- One ACL controls access to the container object itself.
- Another ACL specifies the default ACL on new objects added to the container (the initial object ACL).
- A third ACL specifies the default ACL on new containers added to the container (the initial container ACL).

By default, the `acl` command operates on the ACL of the container object. Use the `-ic` option to operate on the initial container ACL. Use the `-io` option to operate on the initial object ACL. Simple objects, objects that are not container objects, do not have initial container or initial object ACLs.

Some servers that have ACLs also store their network location information in a server entry in the cell directory service (CDS). The server entry has the same name as the server itself and may also have an attached ACL. Use a command option `-entry` to operate on the server entry ACL in CDS rather than the server ACL.

Some DCE objects have more than one purpose. For instance, a registry object can represent a principal and it can also act as a directory or a container. An example is a principal name that identifies another cell (for instance, `/.../friendly.company.com`) with which you want to establish authenticated operation. In this case, our cell maintains a principal name `/.../friendly.company.com`. The registry object for this principal name is:

`/./sec/principal/friendly.company.com`

Let's say that cell also has a hierarchical (subordinate) cell named `/.../friendly.company.com/test_cell`. Our cell maintains another principal name `/.../friendly.company.com/test_cell`. The registry object for this principal name is:

`/./sec/principal/friendly.company.com/test_cell`

Consequently, the registry object `/./sec/principal/friendly.company.com` also acts as a directory because it contains the hierarchical cell name `/./sec/principal/friendly.company.com/test_cell`. The ACL manager that operates on registry objects differs from the ACL manager that operates on registry directories. For instance, the latter ACL manager has an insert (i) permission bit that controls who can add new objects to the directory. Consequently, most `acl` commands provide a `-type` option that lets you specify the appropriate ACL manager when operating on registry objects that are also directories. You can list the ACL managers that are available for registry objects using the `acl show -managers` command.

Attributes
cell default_cellname

Represents the default cell of the ACL. Manipulation of this attribute is possible only through the **modify** and **show** commands.

See the [z/OS DCE Administration Guide](https://www.ibm.com/support/knowledgecenter/EXPRESS/syshelp) for more information about attributes.

**Related Information**

**Commands:**

- dcecp
- group
- organization
- principal
- registry
- rgyedit
- xattrschema
acl check

Returns the permissions granted by the ACL to the principal entering the command.

Format

```
acl check acl_name_list [-entry] [-type acl_manager_name]
```

Options

When no options are specified, the command returns permissions from the ACL for the object named by the command.

- **-entry**
  Specifies that the command operates on the ACL of the name space entry of the object.

- **-type acl_manager_name**
  Specifies that the command uses a particular ACL manager. This option is needed only for objects that have more than one purpose such as principal names that also act as directories (see "Usage" on page 33). List available ACL managers using the `acl show -manager` command.

Usage

The `acl check` operation returns the permissions granted in the specified object's ACL to the principal that started the command. The argument is a list of ACL names being operated on.

**Privilege Required:** You must have `r (read)` permission on the ACL, as determined by the ACL manager.

Examples

dcecp> acl check /.:/hosts
rwtdcia

dcecp> acl check {/./subsys /.:/hosts}
rwtdcia
rwtdcia

dcecp>

Related Information

 Commands:

- `dcecp`
- `group`
- `organization`
- `principal`
- `registry`
- `rgyedit`
- `xattrschema`
acl delete

Deletes all ACL entries from the object, except the user_obj entry.

Format

```
acl delete acl_name_list [-ic | -io | -entry] [-type acl_manager_name]
```

Options

- `-entry` Specifies that the command operates on the ACL of the name space entry of the object.
- `-ic` Specifies that the command operates on the initial container ACL of the named object.
- `-io` Specifies that the command operates on the initial object ACL of the named object.
- `-type acl_manager_name` Specifies that the command uses a particular ACL manager. This option is needed only for objects that have more than one purpose, such as principal names that also act as directories (see “Usage” on page 33). Use the `acl show -manager` command to list available ACL managers.

Usage

The `acl delete` operation removes all ACL entries from the object, except the user_obj entry, if it exists. Not all ACL managers support the user_obj ACL entry. If this command is performed on an ACL that is not capable of having a user_obj ACL entry, or does not have one (resulting in an empty ACL), or has a user_obj ACL entry that does not include c (control) permission, the correct response from the ACL manager is a syntax error. The argument is a list of names of ACLs to be operated on. If an ACL does not exist, an error is generated. An empty string is returned upon a successful completion.

It is easy to lose access to the ACL object if you are not careful. In particular, it is recommended that you do not use `acl delete` against any registry ACL.

Privilege Required: You must have c (control) permission on the ACL, as determined by the ACL manager.

Examples

```
dcecp> acl delete /:/team/myfile
```

Related Information

Commands:

```
dcecp principal rgyedit
group registry xattrschema
organization
```

36 Command Reference
**acl help**

Returns help information about the *acl* object and its operations.

**Format**

```
acl help [operation | -verbose] [-syntax]
```

**Options**

- `-syntax` Displays the syntax diagram of each *acl* operation.
- `-verbose` Displays information about the *acl* object.

**Usage**

The `acl help` command is used without an argument or option to return brief information about each *acl* operation.

Use the optional `operation` argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the `-verbose` option for more detailed information about the *acl* object itself.

Use the `-syntax` option to display the syntax diagram of each operation or of only the operation specified using the `operation` argument.

**Privilege Required:** No special privileges are needed to use the `acl help` command.

**Examples**

```
dcecp> acl help
check
delete              Deletes all ACL entries except 'user_obj' entry if it exists
modify              Adds, removes, or changes ACL entries and attributes.
permissions         Returns permissions associated with an object.
replace             Replaces entire ACL with new ACL entries and attributes.
show                Returns ACL entries or attributes on an object.
help                Prints a summary of command-line options.
operations          Returns the valid operations for command.
dcecp>
```

**Related Information**

Commands:

```
dcecp group organization  principal registry rgynedit xattrschema
```
acl modify

Changes attributes and entries of ACLs.

Format

```
acl modify acl_name_list [-ic | -io | -entry] [-type acl_manager_name]
([-cell new_cell_name] [-purge]
[-add acl_entry_list_with_permissions [-mask {calc | nocalc}]])
[-change acl_entry_list_with_permissions [-mask {calc | nocalc}]])
[-remove acl_entry_list_without_permissions])
```

Options

- **-add acl_entry_list_with_permissions**
  Adds the ACL entries to the ACL. The value of this option is a list of ACL entries with permissions filled in. You can use the -mask option to force or prevent mask recalculation.

- **-cell new_cell_name**
  Changes the value of the cell attribute by specifying the new default cell. It must be the fully qualified name of a single cell, not a list. If the name is not fully qualified, no change is made. The -cell option may be combined with the other options. It is always applied before the other options. Changing the default cell of an ACL that has user or group ACL entries, or their delegate counterparts, may cause problems; the principal and groups mentioned in these ACL entries must be in the default cell. If the default cell changes, these ACL entries must change as well.

- **-change acl_entry_list_with_permissions**
  Changes existing ACL entries in the ACL. The value of this option is a list of ACL entries with permissions filled in. The permissions are the new permissions placed on the specified ACL entries. The ACL entries must exist in the ACL or an error occurs. You can use the -mask option to force or prevent mask recalculation.

- **-entry**
  Specifies that the command operates on the ACL of the name space entry of the object.

- **-ic**
  Specifies that the command operates on the initial container ACL of the named object.

- **-io**
  Specifies that the command operates on the initial object ACL of the named object.

- **-mask**
  If an acl modify operation causes a mask recalculation that unintentionally adds permissions to an existing ACL entry, the modify operation will end with an error unless you specify the -mask option with a value of either calc or nocalc, or a unique abbreviation of one of these values.

  Specifying calc creates or modifies the object's mask_obj type entry with permissions equal to the union of all entries other than type user_obj, other_obj, mask_obj and unauthenticated. Creating or modifying is done after all other modifications to the ACL are performed. The new mask is set even if it grants permissions previously masked out. It is recommended that you use this option only if not specifying it results in an error. If you specify the calc option for an ACL that does not support mask_obj entry type, an error is returned.

  Specifying nocalc means that a new mask will not be calculated.

  This option is only valid if the -add or -change option is present on the modify command. The ACL must support the mask_obj ACL type, or an error is returned. If the user explicitly includes a mask_obj ACL entry in the command (that is, in the value
of the -add or -change options), using the -mask option is not valid. This is because the -mask option resolves unintentional results and the presence of a mask_obj ACL entry implies intent.

Either all changes are made to an individual object, or none are made. Multiple actions can be specified on the command line and processed in a fixed order to guarantee proper processing of the ACL's. See [POSIX.6] for a description of this processing order. The command returns an empty string upon successful completion.

-purge Purges all masked permissions (before any other modifications are made), in all ACL entries except: user_obj, other_obj, mask_obj, user_obj_delegate, other_obj_delegate, and unauthenticated if they exist. This option is useful only for ACL's that contain an entry of type mask_obj.

-remove acl_entry_list_without_permissions
Removes existing ACL entries from the ACL. The value of this option is a list of ACL entries with no permissions. The ACL entries must exist in the ACL or an error occurs. You can use the -mask option to force or prevent mask recalculation.

-type acl_manager_name
Specifies that the command uses a particular ACL manager. This option is needed only for objects that have more than one purpose such as principal names that also act as directories (see "Usage" on page 33). Use the acl show -manager command to list available ACL managers.

Usage

The acl modify operation changes one or more individual ACL entries. The argument is a list of names of ACLs to be modified. They are processed in the order they are entered. The specific operation to perform is described by using options. If multiple options are specified, the order of processing is -cell, -purge, -remove, -change, and -add. To remove the ACL entry for a deleted group or user, first adopt the orphan group or user by entering the group create or principal create command, specifying the orphan's UUID with the uuid option.

Privilege Required: You must have c (control) permission to the ACL being modified as determined by the ACL manager.

Examples

dcecp> acl modify /.:/hosts -add {user melman rwcia}
dcecp> acl modify /.:/hosts -change {user melman rwdtci}

dcecp> acl modify /.:/hosts -add {group dce rwdtcia} \
-remove {user melman}
dcecp>

Related Information

Commands:

dcecp principal rgyedit
group registry xattrschema
organization

Chapter 2. DCE Administration Commands 39
acl operations

Returns a list of the operations supported by the acl object.

Format
acl operations

Usage
The acl operations command uses no arguments, and returns a list of the available operations for the acl object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the acl operations command.

Examples
dcecp> acl operations
check delete modify permissions replace show help operations
dcecp>

Related Information
Commands:
dcecp principal rgyedit
group registry xattrschema
organization
acl permissions

Returns a list describing the permissions associated with an object.

Format

```
acl permissions acl_name_list [-ic | -io | -entry] [-type acl_manager_name]
```

Options

- **-entry** Specifies that the command operates on the ACL of the name space entry of the object.
- **-ic** Specifies that the command operates on the initial container ACL of the named object.
- **-io** Specifies that the command operates on the initial object ACL of the named object.
- **-type acl_manager_name** Specifies that the command uses a particular ACL manager. This option is needed only for objects that have more than one purpose such as principal names that also act as directories (see "Usage" on page 33). Use the acl show -manager command to list available ACL managers.

Usage

The acl permissions operation returns a list describing the permissions associated with an object. The argument is a list of names of ACLs to be operated on. If more than one is entered, the output is concatenated. Each element of the list is a list of two items; the first is the permission token, the second is a description of the permission.

**Privilege Required:** You must have r (read) permission to the ACL, as determined by the ACL manager.

Examples

```
dcecp> acl permissions ./:/hosts
{r {Read entry attributes}}
{w {Update entry attributes}}
{d {Delete entry}}
{t {Test attribute values}}
{c {Change ACL}}
{i {Create new directory entries}}
{a {Administer directory replication}}
```

dcecp>

Related Information

Commands:

```
dcecp principal rgyedit
group registry xattrschema
organization
```
acl replace

Replaces the entire ACL on objects.

Format

```
acl replace acl_name_list -acl acl_entry_list [-ic | -io | -entry] [-type acl_manager_name] [-cell new_default_cellname]
```

Options

- **-acl acl_entry_list**
  Specifies ACL entries and their new values to include in the ACLs being replaced. The syntax of the value of the `-acl` option is a list of ACL entries.

- **-cell new_default_cellname**
  Changes the value of the `cell` attribute by specifying the new default cell. It must be one value, not a list. The `-cell` option may be combined with the other options. It is always applied before the other options.

- **-entry**
  Specifies that the command operates on the ACL of the namespace entry of the object.

- **-ic**
  Specifies that the command operates on the initial container ACL of the named object.

- **-io**
  Specifies that the command operates on the initial object ACL of the named object.

- **-type acl_manager_name**
  Specifies that the command uses a particular ACL manager. This option is needed only for objects that have more than one purpose, such as principal names that also act as directories (see "Usage" on page 33). Use the `acl show -manager` command to list available ACL managers.

Usage

The `acl replace` operation replaces the entire ACL on the object specified by the argument, with the value of the required `-acl` option. The argument is a list of names of ACLs to be operated on. The syntax of the value of the `-acl` option is a list of ACL entries. `acl replace` uses a `-cell` option to specify the new default cell of the ACL. Its value is the name of one cell only (it is not a list). An empty string is returned upon successful completion.

**Privilege Required:** You must have `rwd (read, write, delete)` permission to the ACL, as determined by the ACL manager.

Examples

```
dcecp> acl replace /.:/hosts/hostname/config/epmap \
   -acl {{unauthenticated clidsxt} {any_other clidsxt}}
dcecp>
```
### Related Information

Commands:

<table>
<thead>
<tr>
<th>dcecp</th>
<th>principal</th>
<th>rgyedit</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>registry</td>
<td>xattrschema</td>
</tr>
<tr>
<td>organization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
acl show

Returns a list of the ACL entries for the specified object.

Format

```
acl show acl_name_list [-ic | -io | -entry] [-type acl_manager_name] [-cell] [-managers]
```

Options

- `-cell` Returns the default cell name for the ACL.
- `-entry` Specifies that the command operates on the ACL of the name space entry of the named object.
- `-ic` Specifies that the command operates on the initial container ACL of the named object.
- `-io` Specifies that the command operates on the initial object ACL of the named object.
- `-managers` Returns a list of ACL managers available for the named ACL. This option cannot be specified with the `-type` or `-cell` options.
- `-type acl_manager_name` Specifies that the command uses a particular ACL manager. This option is needed only for objects that have more than one purpose, such as principal names that also act as directories (see "Usage" on page 33). Use the `acl show -managers` command to list available ACL managers.

Usage

The `acl show` operation returns a list of the ACL entries for the specified object. The argument is a list of names of ACLs to be operated on. If more than one is given, the output is concatenated. If they exist, the `mask_obj` and `unauthenticated` ACL entries display first. If the name in an ACL group or user entry cannot obtained (for example, if the group was deleted), then the UUID for the group or principal displays, instead. To use the `acl modify` command to remove the ACL entry for a deleted group or user, first adopt the orphan group or user by entering the `group create` or `principal create`, command specifying the orphan's UUID with the `uuid` option.

Privilege Required: You must have `r` (read) permission to the specified ACLs, as determined by the ACL manager.

Examples

dcecp> acl show /.:/hosts
{unauthenticated r--t---}
{user cell_admin rwdtcia}
{user hosts/absolut/cds-server rwdtcia}
{user hosts/absolut/self rwdtcia}
{user root rwdtcia}
{group subsys/dce/cds-admin rwdtcia}
{group subsys/dce/cds-server rwdtcia}
{any_other r--t---}
dcecp>
Related Information

Commands:

<table>
<thead>
<tr>
<th>dcecp</th>
<th>organization</th>
<th>registry</th>
<th>xattrschema</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>principal</td>
<td>rgyedit</td>
<td></td>
</tr>
</tbody>
</table>
**attrlist**

A dcecp object that manipulates attribute lists.

**Format**

- `attrlist add attrlist -member attrlist`
- `attrlist getvalues attrlist -type type_name`
- `attrlist help [operation | -verbose] [-syntax]`
- `attrlist list attrlist`
- `attrlist modify attrlist`  
  `[-add attribute_type attribute_value...]`
  `[-change attribute_type attribute_value...]`
  `[-remove attribute_type attribute_value...]`
- `attrlist operations`
- `attrlist remove attrlist -member type_name_list`

**Parameters**

- `attrlist`  
  A list of one or more attributes.
  An attribute usually consists of an attribute type and its value. For example:
  `{CDS_Convergence medium}`
  An attribute list contains one or more attributes. For example:
  `{x abc} {CDS_Convergence medium}`

- `operation`  
  The name of one specific attrlist operation (subcommand) about which you want help information.

**Usage**

The attrlist object represents an attribute list as returned or accepted by many dcecp commands. Use this object to check or manipulate attribute lists so that they can be used by other commands, most commonly in scripts.

**Related Information**

Commands:

dcecp
**attrlist add**

Appends one attribute list to another.

**Format**

```plaintext
attrlist add attrlist -member attrlist
```

**Options**

- `member attrlist` Specifies the attribute list to be appended.

**Usage**

The `attrlist add` command returns an attribute list with the attributes specified as the value of the required `-member` option appended.

**Privilege Required:** No special privileges are needed to use the `attrlist add` command.

**Examples**

```plaintext
dcecp> attrlist add {{a b} {c d}} -member {{e f} {g h}}
{a b} {c d} {e f} {g h}
dcecp>
```

**Related Information**

Commands:

```plaintext
dcecp
```
attrlist getvalues

attrlist getvalues

Returns specified attributes from an attribute list.

Format

attrlist getvalues attrlist -type type_name

Options

-type type_name  Specifies the type of the attribute whose value is returned from the attribute list specified.

Usage

The attrlist getvalues command returns the values of all attributes of a type specified by the value of the required -type option. The value may only be a single type, but if the attribute appears more than once in the attribute list, the value of each instance is returned on a separate line.

Privilege Required:  No special privileges are needed to use the attrlist getvalues command.

Examples

dcecp> attrlist getvalues {{a w x} {c y} {a z}} -type a
{w x}
z
dcecp>

This command is likely to be used as commonly as grep to filter the output of show commands. For example:
dcecp> attrlist getvalues [dir show /./hosts] -type CDS_UTS
1996-07-01-10:29:59.265-05:0010.000/00-00-c0-f7-de-56
dcecp>

With abbreviations this could be entered as:
dcecp> at g [dir show ././hosts] -t CDS_UTS
1996-07-01-10:29:59.265-05:0010.000/00-00-c0-f7-de-56
dcecp>

Related Information

Commands:

dcecp
attrlist help

Returns help information about the attrlist object and its operations.

Format

attrlist help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each attrlist operation.
-verbose Displays information about the attrlist object.

Usage

The attrlist help is used without an argument or option to return brief information about each attrlist operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the attrlist object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the attrlist help command.

Examples

dcecp> attrlist help
add Adds attributes to a list.
getvalues Returns the values of specified attributes.
list Returns the attribute types present in a list.
modify Modifies an attribute list.
remove Removes attributes from a list.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp


**attrlist list**

Returns a list of attribute type names from an attribute list.

**Format**

`attrlist list attrlist`

**Usage**

The `attrlist list` command returns a list of all the attribute type names in the attribute list in the order that they appear in the list.

*Privilege Required:* No special privileges are needed to use the `attrlist list` command.

**Examples**

```bash
dcecp> attrlist list {{a b} {c d}}
a c
dcecp>
```

**Related Information**

Commands:

`dcecp`
attrlist modify

Removes and changes attributes and their values in an attribute list.

Format

```
attrlist modify attrlist
    [-add attribute_type attribute_value...] 
    [-change attribute_type attribute_value...] 
    [-remove attribute_type attribute_value...] ]
```

Options

- **-add**
  Specifies new attributes or values to add to existing attributes.

- **-change**
  Specifies to remove all values from an existing attribute and replace with the new values specified.

- **-remove**
  Specifies entire attributes or values to remove.

Usage

The `attrlist modify` command returns an attribute list with attributes modified as specified by the `-add`, `-change`, and `-remove` options. New attributes can be added, or new values added to existing attributes with `-add`. The `-change` option removes all values from an existing attribute and replaces them with new values specified. Entire attributes types or attribute values can be removed with `-remove`. However, you can only specify one value of an attribute you want removed for each type and value pair. If more than one option is specified, the order of operation is `-add`, then `-change`, then `-remove`.

Privilege Required: No special privileges are needed to use the `attrlist modify` command.

Examples

```
dcecp> attrlist modify{{a b} {c d}} -add {{c e}}
{a b} {c d e}
dcecp>

dcecp> attrlist modify {{a b} {c d e}} -remove {{c e}}
{a b} {c d}
dcecp>

dcecp> attrlist modify {{a b} {c d e}} -change {{c f}}
{a b} {c f}
```

Related Information

Commands:

`dcecp`
attrlist operations

attrlist operations

Returns a list of the operations supported by the attrlist object.

Format

attrlist operations

Usage

The attrlist operations command uses no arguments, and returns a list of the available operations for the attrlist object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required:  No special privileges are needed to use the attrlist operations command.

Examples

dcecp> attrlist operations
add getvalues list modify remove help operations

dcecp>

Related Information

Commands:

dcecp
attrlist remove

Removes attributes and their values from an attribute list.

Format

attrlist remove  attrlist -member  type_name_list

Options

-member  type_name_list
    Specifies the attribute types to remove from the attribute list specified.

Usage

The attrlist remove command returns an attribute list after removing attribute types (and their values) specified in the required -member option.

This command removes only entire attributes. To remove specific values, use the attrlist modify command.

Privilege Required:  No special privileges are needed to use the attrlist remove command.

Examples

dcecp>  attrlist remove  {{a b} {c d} {e f} {g h}} -member  {e g}
{a b} {c d}
dcecp>

Related Information

Commands:  
dcecp
A dcecp object that manages the audit daemon on a DCE host.

Format

- **aud disable [audit_daemon_name]**
- **aud enable [audit_daemon_name]**
- **aud help [operation | -verbose] [-syntax]**
- **aud modify [audit_daemon_name] [-change attribute_list | attribute options]**
- **aud operations**
- **aud rewind [audit_daemon_name]**
- **aud show [audit_daemon_name] [-attributes]**
- **aud stop [audit_daemon_name]**

Parameters

- **audit_daemon_name**
  
  By default, operations pertain to the local audit daemon, **audid**. The optional argument **audit_daemon_name** specifies the name or the binding of an audit daemon to operate on. The name syntax is:

  /.../<cellname>/hosts/<hostname>/audit-server

  An audit daemon can also be specified with a string binding for the host on which the audit daemon is running. Use a string binding such as:

  **obj-uuid@ncacn_ip_tcp:130.105.1.227**

  Or, you can specify the binding using a dcecp string syntax such as:

  **{ncacn_ip_tcp:130.105.1.227[1234]}**

  or

  **{ncacn_ip_tcp 130.105.1.227 1234}**

  or

  **{255d2a20-df66-11cd-a9f7-0000c03adff6 ncadg_ip_udp 130.105.5.92}**

- **operation**
  
  The name of one specific **aud** operation (subcommand) about which you want help information.

Usage

The **aud** object represents the audit daemon on a host. The daemon creates audit trails on a single host. The administrative functions are limited to changing the state of the daemon (enabled, disabled, stopped), and changing the strategy used to deal with file system storage for the audit trails.

This command operates on the audit daemon on the local host, unless the _s(aud) convenience variable is set or an argument is specified. If the convenience variable is set or an argument is specified, the value is taken as the name of an audit daemon server entry, as found in the DCE name space, or as a string binding, and that audit daemon is contacted to service the requests. The argument takes precedence over the convenience variable.
Attributes

\texttt{state \{enabled | disabled\}}

Tells whether the audit daemon is accepting audit log requests or not. The values are \texttt{enabled} or \texttt{disabled}. The default value for \texttt{state} is \texttt{enabled}.

\texttt{stostrategy \{save | wrap\}}

The audit trail storage strategy of the daemon. This attribute defines what the daemon does if the audit trail storage is full. Its possible values are:

- \texttt{save}
  If the specified trail size limit is reached (the default is 2 MB), \texttt{auditd} saves the current trail file to a new file (this file has the same name as the original trail file, with the date and time appended). \texttt{auditd} then deletes the contents of the original trail file, and continues auditing from the beginning of this file. This is the default value for \texttt{stostrategy}.

- \texttt{wrap}
  The daemon overwrites the old audit trails.

See the \texttt{z/OS DCE Administration Guide} for more information about audit attributes.

Related Information

Commands:

\texttt{audevents} \hspace{1cm} \texttt{audfilter} \hspace{1cm} \texttt{audtrail} \hspace{1cm} \texttt{dcecp}
aud disable

Disables an audit daemon.

Format

aud disable [audit_daemon_name]

Usage

The aud disable operation disables the audit record logging service of an audit daemon.

The state attribute is changed to disabled. An empty string is returned upon successful completion.

Privilege Required: You must have c (control) permission on the audit daemon ACL. The audit daemon ACL is /.../<cellname>/hosts/<hostname>/audit-server

Examples

dcecp> aud disable
dcecp>

Related Information

Commands:

audevents audfilter audtrail dcecp
aud enable

Enables an audit daemon.

Format

aud enable [audit_daemon_name]

Usage

The aud enable operation enables the audit record logging service of an audit daemon.

The state attribute is changed to enabled. An empty string is returned upon successful completion.

Privilege Required: You must have c (control) permission on the audit daemon ACL. The audit daemon ACL is /.../<cellname>/hosts/<hostname>/audit-server

Examples

dcecp> aud enable
dcecp>

Related Information

Commands:

audevents audfilter audtrail dcecp
aud help

Returns help information about the aud object and its operations.

Format

aud help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each aud operation.
-verbose Displays information about the aud object.

Usage

The aud help command is used without an argument or option to return brief information about each aud operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the aud object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the aud help command.

Examples

dcecp> aud help
disable Disables the audit daemon.
enable Enables the audit daemon.
modify Modifies the attributes of the audit daemon.
rewind Rewinds the specified audit trail file to the beginning.
show Returns the attributes for an audit daemon.
stop Stops the audit daemon.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

audevents audfilter audtrail dcecp
aud modify

Changes the values of audit attributes.

Format

aud modify [audit_daemon_name] {-change attribute_list \ attribute options}

Options

attribute options As an alternative to using the -change option with an attribute list, you can create individual attribute options by adding a hyphen (-) before any attributes listed in the "Attributes" on page 55. The -change option is used in scripts when you can paste in lengthy attribute list output by previous commands. The individual attribute options might be easier to use for interactive commands.

-change attribute_list Lets you specify attributes using an attribute list.

Usage

The aud modify operation allows modification of the audit daemon attributes. aud modify accepts the -change option, which uses an attribute list as a value. It accepts the attribute options -stostrategy and -state. An empty string is returned upon successful completion.

Privilege Required: You must have c (control) permission on the audit daemon ACL. The audit daemon ACL is /.../<cellname>/hosts/<hostname>/audit-server

Examples

dcecp> aud modify -change {{stostrategy wrap} {state enabled}}
dcecp>

dcecp> aud modify -stostrategy wrap -state enabled
dcecp>

Related Information

Commands:

audevents audfilter audtrail dcecp
aud operations

Returns a list of the operations supported by the aud object.

**Format**

aud operations

**Usage**

The **aud operations** command uses no arguments and returns a list of the available operations for the aud object. The order of the elements is alphabetic, except that help and operations are listed last.

**Privilege Required:** No special privileges are needed to use the **aud operations** command.

**Examples**

```
dcecp> aud operations
disable enable modify rewind show stop help operations
```

**Related Information**

Commands:

audevents audfilter audtrail dcecp
aud rewind

Rewinds the central audit trail file to the beginning.

Format

```
aud rewind [audit_daemon_name]
```

Usage

The central trail file is rewound to the beginning. An empty string is returned upon successful completion.

Privilege Required: You must have **c (control)** permission on the audit daemon ACL. The audit daemon ACL is `/.../<cellname>/hosts/<hostname>/audit-server`

Examples

```
dcecp> aud rewind
dcecp>
```

Related Information

Commands:

```
audevents       audfilter       audtrail       dcecp
```
aud show

Returns the attribute list for the audit daemon.

Format

`aud show [audit_daemon_name] [-attributes]`

Usage

The `aud show` operation returns the attribute list for the audit daemon. The `-attributes` option is provided for consistency with other `dcecp` commands; it does not change the performance of the command.

Privilege Required: You must have `r (read)` permission on the audit daemon ACL. The audit daemon ACL is `/.../<cellname>/hosts/<hostname>/audit-server`

Examples

```
dcecp> aud show
{state enabled}
{stostrategy wrap}
dcecp>
```

Related Information

Commands:

- `audevents`
- `audfilter`
- `audtrail`
- `dcecp`
**aud stop**

Stops the audit daemon.

**Format**

`aud stop [audit_daemon_name]`

**Usage**

The `aud stop` operation stops the audit daemon process. An empty string is returned upon successful completion.

**Privilege Required:** You must have c (control) permission on the audit daemon ACL. The audit daemon ACL is `/.../<cellname>/hosts/<hostname>/audit-server`

**Examples**

```
dcecp> aud stop
dcecp>
```

**Related Information**

Commands:

```
audevents audtrail dcecp
audfilter
```
audevents

A **dcecp** object that lists audit events on a DCE host.

**Format**

```
audevents catalog
audevents help [operation | -verbose] [-syntax]
audevents operations
audevents show event_class_list
```

**Parameters**

*event_class_list*  A list of names of event classes. An event class is a group of audit events. Event classes are defined in event class files created in the /opt/dcelocal/etc/audit/ec directory. See the **z/OS DCE Administration Guide** for more information.

*operation*  The name of one specific **audevents** operation (subcommand) about which you want help information.

**Usage**

This object represents the event classes that are recognized by an audit daemon on a host. Each event class is defined in an event class configuration file, and the file name is the symbolic name of the event class.

This command operates on the audit daemon on the local host.

**Related Information**

Commands:

```
aud audfilter audtrail dcecp
```
**audevents catalog**

Returns a list of the names of all event classes.

**Format**

`audevents catalog`

**Usage**

The **audevents catalog** operation returns a list of the names of all event classes. This command takes no arguments. The order returned is arbitrary.

**Privilege Required:** To run **audevents catalog**, you need **read** permission to the files in the event class directory, `/opt/dcelocal/etc/audit/ec`

**Examples**

```
dcecp> audevents catalog
dce_audit_admin_modify
dce_audit_admin_query
dce_audit_filter_modify
dce_audit_filter_query
dce_dts_mgt_modify
dce_dts_mgt_query
dce_dts_synch
dce_dts_time_provider
dce_sec_authent
dce_sec_control
dce_sec_modify
dce_sec_query
dce_sec_server
dcecp>
```

**Related Information**

Commands:

```
aud audfilter audtrail audtrail dcecp
```
audevents help

Returns help information about the audevents object and its operations.

Format

audevents help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each audevents operation.
-verbose Displays information about the audevents object.

Usage

The audevents help command is used without an argument or option to return brief information about each audevent operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the audevents object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the audevents help command.

Examples

dcecp> audevents help
catalog Returns the list of event classes for an audit daemon.
show Returns the contents of an event class.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

aud audfilter audtrail dcecp
audevents operations

Returns a list of the operations supported by the audevents object.

Format

audevents operations

Usage

The audevents operations command uses no arguments and returns a list of the available operations for the audevents object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the audevents operations command.

Examples

dcecp> audevents operations
catalog show help operations
dcecp>

Related Information

Commands:

aud audfilter audtrail dcecp
audevents show

Returns the contents of an event class.

Format

audevents show event_class_list

Usage

The audevents show operation returns the contents of an event class. The argument is a list of names of event classes. For each event class, audevents show returns a list consisting of one name and several 32 bit integers displayed in hexadecimal format. The name is the event class name, the rest are the numbers of the events that are in the event class. If more than one event class is given in the argument, the output list of each class is returned.

Privilege Required: You must have read permission to the files in directory /opt/dcelocal/etc/audit/ec

Examples

dcecp> audevents show dce_audit_filter_query
{dce_audit_filter_query 0x301 0x302}
dcecp>

dcecp> audevents show {dce_audit_filter_query dce_dts_time_provider}
{dce_audit_filter_query 0x301 0x302}
{dce_dts_time_provider 0x211 0x212}
dcecp>

Related Information

Commands:

aud audfilter audtrail dcecp
audfilter

A **dcecp** object that manages the event filters on a DCE host.

**Format**

- `audfilter catalog`
- `audfilter create audit_filter_name_list -attribute guide_name_list`
- `audfilter delete audit_filter_name_list`
- `audfilter help [operation | -verbose] [-syntax]`
- `audfilter modify audit_filter_name_list [|-add guide_name_list] [|-remove guide_name_list]`
- `audfilter operations`
- `audfilter show audit_filter_name_list`

**Parameters**

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>audit_filter_name_list</code></td>
<td>A list of one or more audit event filter names. A filter name consists of a filter type and possibly a key, depending on the type.</td>
</tr>
</tbody>
</table>

The audit filter types are as follows:

- **cell**
  - The key is a `cell_name`.

- **cell_overridable**
  - The key is a `cell_name`.

- **foreign_group**
  - The key is a `.../cellname/group_name`.

- **foreign_principal**
  - The key is a `.../cellname/principal_name`.

- **group**
  - The key is a `group_name`.

- **principal**
  - The key is a `principal_name`.

- **world**
  - This type has no key.

- **world_overridable**
  - This type has no key.

Examples of audit filter names are: `principal melman`, `group dce`, and `world`.

- **operation**
  - The name of one specific **audfilter** operation (subcommand) about which you want to see help information.

**Data Structures**

Several **audfilter** operations add and remove guide data that is stored in a filter. A guide specifies the actions to take when a particular audit condition occurs. A single filter can contain multiple guides specifying various actions for different conditions. A guide is identified by a list of the three elements that make up the guide:

- audit actions
- audit conditions
- event classes

A guide specifies what (event classes), when (audit conditions), and how (audit actions) to audit. For further information see the [z/OS DCE Administration Guide](zos_dce_admin_guide).
Event classes are definable by the administrator.

**Audit Actions**

<table>
<thead>
<tr>
<th>Audit Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alarm</td>
<td>Sends the audit record to the system console.</td>
</tr>
<tr>
<td>all</td>
<td>Specifies to log the event and signal the alarm. If all is set as a guide action, the <code>audfilter show</code> commands return the action all, not {log alarm all}.</td>
</tr>
<tr>
<td>log</td>
<td>Logs the audit record either in the audit trail file of the audit daemon or a user-specified audit trail file.</td>
</tr>
<tr>
<td>none</td>
<td>This command uses no audit action.</td>
</tr>
</tbody>
</table>

**Audit Conditions**

<table>
<thead>
<tr>
<th>Audit Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>denial</td>
<td>Audit only if the event failed because of access denials.</td>
</tr>
<tr>
<td>failure</td>
<td>Audit only if the event failed because of other reasons.</td>
</tr>
<tr>
<td>pending</td>
<td>The outcome was not determined.</td>
</tr>
<tr>
<td>success</td>
<td>Audit only if the event succeeded.</td>
</tr>
</tbody>
</table>

**Usage**

The `audfilter` object represents audit event filters that are kept by the audit daemon to determine if an auditable event is logged. An audit filter consists of a list of guides. The name of the audit filter is a filter type and possibly a key (depending on the type).

This command operates on the audit daemon on the local host, unless the `_s(aud)` convenience variable is set. If set, the value is taken as the name of an audit daemon's server entry, as found in the DCE name space, or as a string binding; and that audit daemon is contacted to service the requests.

**Related Information**

Commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aud</td>
<td></td>
</tr>
<tr>
<td>audevents</td>
<td></td>
</tr>
<tr>
<td>audtrail</td>
<td></td>
</tr>
<tr>
<td>dcecp</td>
<td></td>
</tr>
</tbody>
</table>
audfilter catalog

Returns a list of names of all filters known to the audit daemon.

Format

audfilter catalog

Usage

The audfilter catalog operation returns a list of filter names maintained by the audit daemon. This command uses no arguments. The names consist of a list of filter types and if necessary a key. They are returned in an arbitrary order.

Privilege Required: You must have r (read) permission on the audit daemon's ACL.

Examples

The following example returns the names of four filters from the audit daemon:

dcecp> audfilter catalog
{principal melman}
{foreign_principal /.../cell_X/kevins}
{group dce}
world
dcecp>

Related Information

Commands:

aud audevents audtrail dcecp
audfilter create

audfilter create

Creates a new audit filter.

Format

`audfilter create audit_filter_name_list -attribute guide_name_list`

Options

-attribute `guide_name_list`

Specifies a list of one or more guides being added to the specified audit event filters that are created. A guide name consists of three elements; an event class, an audit condition, and an audit action. See "Data Structures" on page 69 for information about guide names.

Usage

The `audfilter create` operation creates a new audit filter. The argument is a list of names of audit filters being created. Because a filter that has no guides is removed by the audit daemon during a clean-up phase, this command requires an -attribute option whose value is a list of guides being added to the specified audit filters on creation. All guides are added to all audit filters specified being created. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission on the audit daemon's ACL.

Examples

```bash
  dcecp> audfilter create {principal melman} -attribute {dce_sec_query denial log}
  dcecp>
```

Related Information

Commands:

`aud` `audevents` `audittrail` `dcecp`
audfilter delete

Deletes the filter including all filter guides.

Format

audfilter delete audit_filter_name_list

Usage

The audfilter delete operation deletes the filter including all filter guides. The argument is a list of names of audit filters being deleted. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission on the audit daemon's ACL.

Examples

dcecp> audfilter delete {principal melman}
dcecp>

Related Information

Commands:

aud audevents audittrail dcecp
audfilter help

Displays help information about the audfilter object and its operations.

Format

audfilter help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each audfilter operation.
-verbose Displays detailed information about the audfilter object.

Usage

The audfilter help command is used without an argument or option to return brief information about each audfilter operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the audfilter object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required No special privileges are needed to use the audfilter help command.

Examples

dcecp> audfilter help
catalog Returns the list of filters for an audit daemon.
create Creates a new filter with specified guides.
delete Deletes a filter and its associated guides.
modify Adds or removes one or more guides of a filter.
show Returns a list of guides in a specified filter.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

aud audevents audtrail dcecp
audfilter modify

Adds or removes one or more guides of a filter.

Usage

audfilter modify audit_filter_name_list [-add guide_name_list] [-remove guide_name_list]

Options

- add guide_name_list
  Specifies a list of one or more guides being added to the specified audit event filters that are being modified. A guide name consists of three elements; an audit condition, an audit action, and an event class. See “Data Structures” on page 69 for information about guide names.

- remove guide_name_list
  Specifies a list of one or more guides being removed from the specified audit event filters that are being modified. A guide name consists of three elements; an audit condition, an audit action, and an event class. See “Data Structures” on page 69 for information about guide names.

Usage

The audfilter modify operation adds or removes one or more guides of a filter. The argument is a list of names of audit filters being modified. In addition, the specific operation to perform is described with one or more of the following options: -add and -remove. The argument for both of these options is a list of guides. If more than one guide is specified, all guides are operated on, but not atomically. If the last guide is removed from a filter, the filter is deleted at some point by the audit daemon. The guides specified with -add are added before the guides specified with -remove are removed.

Atomicity of multiple actions is not guaranteed. Similarly, the effect of adding a guide that partially exists in the specified filter is to change the existing guides. These changes guarantee that the semantics of the removal or addition is maintained.

Privilege Required: You must have w (write) permission on the audit daemon's ACL.

Examples

dcecp> audfilter modify {principal rousseau} -add \
   {dce_dts_mgt_modify failure alarm} -remove \
   {dce_dts_mgt_query all log}
dcecp>

Related Information

Commands:

aud    audedeuals    audtrail    dcecp
audfilter operations

Returns a list of the operations supported by the audfilter object.

Format

audfilter operations

Usage

The audfilter operations command uses no arguments, and returns a list of the available operations for the audfilter object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the audfilter operations command.

Examples

dcecp> audfilter operations
   catalog create delete modify show help operations
   dcecp>

Related Information

Commands:

  aud         audevents         audtrail         dcecp
audfilter show

Returns a list of guides in a specified filter.

Format

audfilter show audit_filter_name_list

Usage

The audfilter show operation returns a list of guides in a specified filter. The argument is a list of filter names (a filter type, and if needed a key) being shown. If more than one is entered, the output is concatenated with a blank line between filters.

Privilege Required: You must have r (read) permission on the audit daemon’s ACL.

Examples

dcecp> audfilter show {principal rousseau}
{dce_dts_mgt_modify failure alarm}
{dce_dts_mgt_query all log}
dcecp>

Related Information

Commands:

aud audevents audtrail dcecp
Audtrail

A dcecp object that converts the audit trail into a human-readable format.

Format

audtrail help [operation | -verbose] [-syntax]
audtrail operations
audtrail show audit_trail_file_name_list [to filename]

Parameters

audit_trail_file_name_list
This argument is a list of one or more names of audit trail files.

operation
The name of one specific audevents operation (subcommand) about which you want help information.

Usage

The audtrail object represents an audit trail file. This object supports only one operation that converts the audit trail into a human readable format.

Related Information

Commands:

aud audevents audfilter dcecp
audtrail help

Returns help information about the audtrail object and its operations.

Format

audtrail help [operation | -verbose] [-syntax]

Options

  -syntax  Displays the syntax diagram of each audtrail operation.
  -verbose Displays information about the audtrail object.

Usage

The audtrail help command is used without an argument or option to return brief information about each audtrail operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the audtrail object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required:  No special privileges are needed to use the audtrail help command.

Examples

dcecp> audtrail help

    show    Returns or files the contents of an audit trail file.
    help    Prints a summary of command-line options.
    operations Returns the valid operations for command.

dcecp>

Related Information

Commands:

aud    audevents    audfilter    dcecp
audtrail operations

Returns a list of the operations supported by the **audtrail** object.

**Format**

`audtrail operations`

**Usage**

The **audtrail operations** command uses no arguments, and returns a list of the available operations for the **audtrail** object. The order of the elements is alphabetic, except that **help** and **operations** are listed last.

**Privilege Required:** No special privileges are needed to use the **audtrail operations** command.

**Examples**

```
dcecp> audtrail operations
show help operations
dcecp>
```

**Related Information**

Commands:

- `aud`
- `audevents`
- `audfilter`
- `dcecp`
audtrail show

Returns the audit trail in a human readable format.

Format

audtrail show audit_trail_file_name_list [-to filename]

Options

- `-to filename` Specifies the destination file name for the trail. If none is specified, the trail is returned from the command. If specified, the command returns an empty string.

Usage

The `audtrail show` operation returns the audit trail in a human readable format. This command uses a list of names of audit trail files as an argument. If more than one name is given, then the output of each audit trail is concatenated with a blank line between audit trails. The `-to` option can be used to specify a destination file name for the trail.

Because audit trail files may grow quite large, it is recommended to use the `-to` switch to avoid reading the entire trail into memory.

Privilege Required: You must have read permission on the audit trail file on the local file system.

Examples

The following command shows the contents of the audit trail file `my_trail`:

dcecp> audtrail show my_trail

--- Start of an event record --- Event Number: 259
Client: /.../stp.good.com/hosts/drinkernisti/self
Event Outcome: success
Authorization Status: Authorized with a pac
Local Time: 1996-12-19-19:02:27.037-05:00I-----
--- End of an event record ---

--- Start of an event record --- Event Number: 256
Client: /.../stp.good.com/hosts/drinkernisti/self
Event Outcome: success
Authorization Status: Authorized with a pac
Local Time: 1996-12-19-19:02:28.819-05:00I-----
--- End of an event record ---

dcecp>

Related Information

Commands:

aud audevents audfilter dcecp
A dcecp object that manages a cdscache.

**Format**

```
cdscache create cache_name -binding server_binding
cdscache delete cache_name_list
cdscache dump
cdscache help [operation | -verbose] [-syntax]
```

```
cdscache operations
```

```
cdscache show server_name_list {-server | -clearinghouse}
```

**Parameters**

- `cache_name`: A simple name for the cached server machine. A simple name is like `pelican`; it is not a cell-relative name like `/.:/hosts/pelican`.
- `cache_name_list`: A list of one or more cache names.
- `operation`: The name of one specific cdscache operation (subcommand) about which you want help information.
- `server_name_list`: A list of one or more simple names of servers or CDS names of clearinghouses.

**Usage**

The cdscache object represents the CDS cache on the local node. The CDS cache contains information about servers and clearinghouses known to the local machine, and also contains user data about CDS entries that were read. The create and delete operations apply only to the server information. The show and dump commands can display additional information.

**Related Information**

Commands:

```
cdscp dcecp link
clearinghouse directory object
```
**cdscache create**

Creates knowledge of a server in the local clerk's cache.

**Format**

```
cdscache create cache_name -binding server_binding
```

**Options**

- **-binding** *server_binding*
  
The required `-binding` option lets you specify the binding information for a CDS server. This option uses a *server_binding* argument which is the protocol sequence and network address of the server node. The string format is:

  ```
  protocol-sequence:network-address
  ```

  The TCL format is:

  ```
  {protocol-sequence network-address}.
  ```

  A *protocol-sequence* is a character string identifying the network protocols used to establish a relationship between a client and server. The protocol sequence has a specific format that depends on the network address that is supplied in the binding; for example, `ncacn_ip_tcp` or `ncadg_ip_udp`. For *network-address*, specify a string representing the network address of the server node; for example, an Internet address using the common Internet address notation. For more information about this format, see the RPC introduction in the [z/OS DCE Application Development Reference](https://www.ibm.com/support/knowledgecenter/SSEPGG_1.3.0/com.ibm.zos.dce.doc/).

**Usage**

The `cdscache create` operation creates knowledge of a server in the local clerk's cache. The *cache_name* argument is a simple name for the cached server. A simple name is like `pelican`; it is not a cell-relative name like `/.:/hosts/pelican`. This command is typically used to manually provide configuration information to a clerk that cannot automatically configure itself. This is required, for instance, to give the clerk addressing information about a server across a WAN. When the clerk knows about one server, it can find other servers through referrals. The command returns an empty string on success.

**Privilege Required:** You must have *w* (write) permission to the clerk, 

`/.../cellname/hosts/hostname/cds_clerk`

**Examples**

The following command creates knowledge of the server `pelican` in the local clerk's cache:

```
dcecp> cdscache create pelican -binding ncacn_ip_tcp:16.20.15.25
```

**Note:** The address used must be a valid address in the cell.

**Related Information**

Commands:

- `cdscp`
- `clearinghouse`
- `dcecp`

Chapter 2. DCE Administration Commands 83
cdscache create

directory   link   object
cdscache delete

Removes knowledge of a server that you specifically created from the local clerk's cache.

Format

\texttt{cdscache delete cache\_name\_list}

Usage

The \texttt{cdscache delete} operation removes knowledge of a server that you specifically created from the local clerk's cache. The required \texttt{cache\_name\_list} argument is a list of simple names given with the \texttt{cdscache create} command. You can only delete a server that you have specifically created with the \texttt{cdscache create} command.

Privilege Required: You must have \texttt{w (write)} permission to the clerk, \\
\texttt{/.../cellname/hosts/hostname/cds\_clerk}

Examples

The following command removes knowledge of the server \texttt{pelican} from the clerk cache:
\begin{verbatim}
dcecp> cdscache delete pelican
\end{verbatim}

Related Information

Commands:

\begin{verbatim}
cdscp
clearinghouse
dcecp
directory
link
object
\end{verbatim}
cdscache dump

discache dump

Displays the entire contents of the clerk cache.

Format

\texttt{cdscache dump}

Usage

The \texttt{cdscache dump} operation displays the contents of the clerk cache on the screen.

Privilege Required: You must have superuser or root \texttt{uid=0} authority on the clerk system.

Examples

The following command displays the contents of the clerk cache on the screen:

\begin{verbatim}
dcecp> cdscache dump
CACHED DIRECTORIES*************************************************************************
/.../dcecell22.endicott.ibm.com/hosts/dcecell22
  Master at /.../dcecell22.endicott.ibm.com/dcecell22_ch
  Clearinghouse UID: 03ca0d02-29d8-11cf-a11c-08005a191ee8
  Clearinghouse Address:
     ncacn_ip_tcp:9.130.79.119[]
     ncadg_ip_udp:9.130.79.119[]

/.../dcecell22.endicott.ibm.com/subsys/dce/pwd_mgmt
  Master at /.../dcecell22.endicott.ibm.com/dcecell22_ch
  Clearinghouse UID: 03ca0d02-29d8-11cf-a11c-08005a191ee8
  Clearinghouse Address:
     ncacn_ip_tcp:9.130.79.119[]
     ncadg_ip_udp:9.130.79.119[]

...
...
...

dcecp>
\end{verbatim}

Related Information

Commands:

\texttt{cdscp} \hspace{1cm} \texttt{dcecp} \hspace{1cm} \texttt{link}
\texttt{clearinghouse} \hspace{1cm} \texttt{directory} \hspace{1cm} \texttt{object}
cdscache help

Returns help information about the cdscache object and its operations.

Format

`cdscache help [operation | -verbose] [-syntax]`

Options

- `-syntax` Displays the syntax diagram of each cdscache operation.
- `-verbose` Displays information about the cdscache object.

Usage

The `cdscache help` command is used without an argument or option to return brief information about each cdscache operation.

Use the optional `operation` argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the `-verbose` option for more detailed information about the cdscache object itself.

Use the `-syntax` option to display the syntax diagram of each operation or of only the operation specified using the `operation` argument.

Privilege Required: No special privileges are needed to use the cdscache help command.

Examples

```
dcecp> cdscache help
create          Adds information about named server in local cds cache.
delete          Removes information about named server in local cds cache.
dump            Dump all information from local cds cache.
show            Returns information stored in cds cache.
help            Prints a summary of command-line options.
operations      Returns the valid operations for command.
dcecp>
```

Related Information

 Commands:

<table>
<thead>
<tr>
<th>cdscp</th>
<th>dcecp</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>clearinghouse</td>
<td>directory</td>
<td>object</td>
</tr>
</tbody>
</table>
cdscache operations

Returns a list of the operations supported by the cdscache object.

Format

cdscache operations

Usage

The cdscache operations command uses no arguments, and returns a list of the available operations for the cdscache object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the cdscache operations command.

Examples

dcecp> cdscache operations
create delete dump show help operations
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscp</th>
<th>dcecp</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>clearinghouse</td>
<td>directory</td>
<td>object</td>
</tr>
</tbody>
</table>
cdscache show

Displays information about clearinghouses or servers stored in the cache.

Format

```
cdscache show server_name_list (-clearinghouse | -server)
```

Options

- **-clearinghouse**
  The `-clearinghouse` option displays all the names and values of the attributes in the specified cached clearinghouse. The following are valid attributes:

  **Creation Time**
  Specifies the time at which this clearinghouse was added to the cache.

  **Miscellaneous Operations**
  Specifies the number of operations other than read and write (that is, skulks, new epochs, and so on) performed by this clerk on the cached clearinghouse.

  **Read Operations**
  Specifies the number of lookup operations of any sort performed by the clerk on the cached clearinghouse.

  **Towers**
  Specifies the protocol sequence and Internet address of the server that maintains the cached clearinghouse.

  **Write Operations**
  Specifies the number of write operations performed by this clerk on the cached clearinghouse.

- **-server**
  The `-server` option displays address information of a server in the local clerk’s cache. The following list describes the valid attributes:

  **Name**
  The directory cell name

  **Towers**
  The protocol sequence and network address of the server node

Usage

The cdscache show operation displays information about clearinghouses or servers stored in the cache. The required `server_name_list` argument is a list of one or more simple names of servers or CDS names of clearinghouses for which you want to display information. You must use either the `-clearinghouse` or `-server` option to select the information you want to display.

Privilege Required: You must have `r (read)` permission to the clerk,

```
/.../cellname/hosts/hostname/cds_clerk
```

Examples

The following command displays all attributes of the cached clearinghouse `./:claire_ch`. 
cdscache show

dcecp> cdscache show /./claire_ch -clearinghouse
(CH_Name /.../blue.cell.osf.org/claire_ch)
(Created 1996-10-07-11:41:23.131)
Others 458
Reads 150221
(Tower {ncacn_ip_tcp 130.105.4.158})
(Tower {ncadg_ip_udp 130.105.4.158})
Writes 162

dcecp>

The following command displays all attributes of the cached server drkstr.

dcecp> cdscache show drkstr -server
(CH_Name /.../terrapin_cell.osf.org/drkstr_ch)
(Tower {ncacn_ip_tcp 130.105.5.16})
(Tower {ncadg_ip_udp 130.105.5.16})

dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscp</th>
<th>dcecp</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>clearinghouse</td>
<td>directory</td>
<td>object</td>
</tr>
</tbody>
</table>
cell

A dcecp task object that operates on a DCE cell.

Format

    cell help [operation] [-verbose] [-syntax]

cell operations

    cell ping [cell_name] [-replicas] [-clients]

cell show [cell_name] [-simplename]

Parameters

    cell_name     This optional argument is the name of a single cell to operate on. The name must be a fully qualified cell name as in:
                  /.../their_cell.goodco.com

    operation     The name of one specific cell operation (subcommand) about which you want help information.

Usage

The cell object represents a single DCE cell as a whole. This includes all machines, services, resources, and principals. The optional argument to the cell object is a single cell name (not a list of cell names). If omitted, /.: is the default.

Attributes

The cell show command returns the following attributes and one or more values:

    cdsservers    Each value is the name of a machine running a CDS server. The name is the simple name found under /.:/hosts.

    dtsservers    Each value is the name of a DTS server in the cell.

    hosts         Each value is the name of a host in the cell, including machines mentioned above as servers. For example:
                   hosts/machine1.

    secservers    Each value is the name of a security server.

See the z/OS DCE Administration Guide for more information about attributes.

Related Information

Commands:

dcecp directory host server
cell help

Returns help information about the cell object and its operations.

Format

cell help [operation | -verbose] [-syntax]

Options

-verbose Displays information about the cell object.
-syntax Displays the syntax diagram of each cell operation.

Usage

The cell help command is used without an argument or option to return brief information about each cell operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the cell object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the cell help command.

Examples

dcecp> cell help
ping Shows the current server status of the cell.
show Shows attributes describing the configuration of a cell.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp directory host server
cell operations

Returns a list of the operations supported by the cell object.

Format

cell operations

Usage

The cell operations command uses no arguments, and returns a list of the available operations for the cell object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the cell operations command.

Examples

dcecp> cell operations
  ping show help operations
  dcecp>

Related Information

Commands:

dcecp directory host server
cell ping

Performs quick checks to test whether a cell is running.

Format

cell ping [cell_name] [-replicas] [-client]

Options

- **-clients**
  This option causes the command to ping every machine in the cell. It does this by looping through /./hosts and doing a host ping on each host name. In case of failure, it generates an error and returns a list of hosts that cannot be contacted. When successful, it returns the message DCE clients available.

- **-replicas**
  This option causes the command to ping each security and CDS replica. In case of failure, it generates an error and returns a list of servers that cannot be contacted. When successful, it returns the message DCE servers available.

Note: When specifying both -replicas and -clients, this command returns DCE clients and servers available when successful.

Usage

If called with no option, the cell ping operation pings (using server ping) the master security server, the CDS server that has the master root replica, and all the DTS servers in the cell. In case of failure, it generates an error and returns a list of servers that cannot be contacted. When successful, it returns the message DCE services available.

Privilege Required: You must have r (read) permission to several directories, /./hosts, /./hosts/hostname, and /./sybsys/dce/sec/master, in the CDS name space.

Examples

The following example tells whether the core services master servers are available:

dcecp> cell ping /.../blue.cell.osf.org
DCE services available
dcecp>

The following example tells whether the core services and their replicas are available:

dcecp> cell ping -replicas
DCE servers available
dcecp>

The following example tests the presence of all DCE hosts in a cell:

dcecp> cell ping -clients
DCE clients available
dcecp>

Related Information

Commands:
<table>
<thead>
<tr>
<th>dcecp</th>
<th>directory</th>
<th>host</th>
<th>server</th>
</tr>
</thead>
</table>

cell ping
cell show

Returns attributes describing the configuration of the specified cell.

Format

cell show [cell_name] [-simplename]

Options

-simplename Specifies that the names of servers in the cell are displayed as cell relative, not fully qualified.

Usage

The cell show operation returns attributes describing the configuration of the specified cell. The returned attributes are:

- cdsservers Each value is the name of a machine running a CDS server.
- dtsservers Each value is the name of a DTS server in the cell.
- hosts Each value is the name of a host in the cell, including machines mentioned above as servers. For example:
  {hosts/machine1}
- secservers Each value is the name of a security server.

See the z/OS DCE Administration Guide for more information about attributes.

Privilege Required: You must have r (read) permission to several directories in the CDS name space.

Examples

dcecp> cell show /.../dcecp.cell.osf.org
 {secservers
  /.../dcecp.cell.osf.org/subsys/dce/sec/ice
  /.../dcecp.cell.osf.org/subsys/dce/sec/master}
 {cdsservers
  /.../dcecp.cell.osf.org/hosts/frick}
 {dtsservers
  /.../dcecp.cell.osf.org/hosts/frick
  /.../dcecp.cell.osf.org/hosts/ice
  /.../dcecp.cell.osf.org/hosts/ninja}
 {hosts
  /.../dcecp.cell.osf.org/hosts/frick
  /.../dcecp.cell.osf.org/hosts/ice
  /.../dcecp.cell.osf.org/hosts/ninja}
dcecp>
Related Information

Commands:

dcecp  directory  host  server
clearinghouse

A dcecp object that manages a clearinghouse in the DCE Cell Directory Service.

Format

```
clearinghouse catalog [cell_name_list] [-simplename]
clearinghouse create clearinghouse_name_list
clearinghouse delete clearinghouse_name_list
clearinghouse disable clearinghouse_name_list
clearinghouse help [operation] [-verbose] [-syntax]
clearinghouse initiate clearinghouse_name_list -checkpoint
clearinghouse operations
clearinghouse repair clearinghouse_name_list -timestamps
clearinghouse show clearinghouse_name_list [-counters] [-attribute] [-all] [-schema]
clearinghouse verify clearinghouse_name_list
```

Parameters

- **cell_name_list**: This optional argument is a list of one or more cell names. Each name must be a fully qualified cell name as in:
  ```
  /.../their_cell.goodco.com
  ```

- **clearinghouse_name_list**: A list of one or more names of clearinghouses on which you want to operate.

- **operation**: The name of one specific clearinghouse operation (subcommand) about which you want help information.

Usage

The clearinghouse object represents CDS clearinghouses. Clearinghouses are databases located on CDS server machines that store data (directories, objects, and links) in the CDS. Files on the server machines contain the actual clearinghouse data. Clearinghouses are also represented in the CDS name space by an entry that contains information about the clearinghouse.

You must run the **create** command on the host where you want to create the new clearinghouse and the **delete**, **disable**, **initiate**, **repair**, and **verify** commands on the host where the clearinghouse to be operated on resides.

If the **_s(cds)** convenience variable is set, it is treated as the name of a clearinghouse to contact for this operation. This is the only clearinghouse that is contacted to complete the operation. These commands do not set the value of this variable after completion.

Attributes

The following are the CDS-defined attributes that may be in CDS clearinghouse objects:

- **CDS_AllUpTo**: Specifies the date and time the clearinghouse object was updated to reflect the **CDS_CHDirectories** attribute.
CDS_CHDirectories
Specifies the full name and unique identifier (UUID) of every directory that has a replica in this clearinghouse.

CDS_CHLastAddress
Specifies the current reported network address of the clearinghouse.

CDS_CHName
Specifies the full name of the clearinghouse.

CDS_CHState
Specifies the state of the clearinghouse. The state on specifies the clearinghouse is running and available.

CDS_CTS
Specifies the creation timestamp (CTS) of the clearinghouse.

CDS_DirectoryVersion
Specifies the current version of the directory in the clearinghouse in which the directory was created.

CDS_NSCellname
Specifies the name of the cell in which the clearinghouse resides.

CDS_ObjectUUID
Specifies the unique identifier of the clearinghouse.

CDS_ReplicaVersion
Specifies the current version of the replica in which the directory was created.

CDS_UpgradeTo
A single-valued attribute used to control the upgrading of a directory from one version of CDS to another. By modifying this attribute, the process of upgrading a directory to a newer version of CDS may begin.

CDS_UTS
Specifies the timestamp of the most recent update to an attribute of the clearinghouse.

Counters
The clearinghouse counters are:

missingentries
Specifies the number of times the clearinghouse entry missing event was generated.

corruptions
Specifies the number of times that the data corruption event was generated.

skulkfailures
Specifies the number of times that a skulk of a directory, started from this clearinghouse, failed to complete (usually because one of the replicas in the replica set was unreachable).

failedupgrades
Specifies the number of times that upgrades failed.

disables
Specifies the number of times that the clearinghouse was disabled since it was last started.

enables
Specifies the number of times that the clearinghouse was enabled since it was last started.

returnedrefs
Specifies the number of requests directed to this clearinghouse that resulted in the return of a partial answer instead of satisfying the client's request.

reads
Specifies the number of read operations directed to this clearinghouse.

rootunreachables
Specifies the number of times the root lost event was generated.

writes
Specifies the number of write operations directed to this clearinghouse.

See the [z/OS DCE Administration Guide](https://www.ibm.com) for more information about attributes and counters.
Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>dcecp</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>directory</td>
<td>object</td>
</tr>
</tbody>
</table>
clearinghouse catalog

Returns a list of the names of all clearinghouses in a cell.

Format

clearinghouse catalog [cell_name_list] [-simplename]

Options

-simplename  Specifies that the names of clearinghouses are displayed as cell-relative, not fully qualified.

Usage

The clearinghouse catalog command returns a list of the names of all clearinghouses in a cell. If you do not specify the optional argument, the cell name defaults to ./.

Privilege Required:  No special privileges are needed to use the clearinghouse catalog command.

Examples

dcecp> clearinghouse catalog
./dcecp.cell.osf.org/frick_ch

dcecp>

Related Information

Commands:

cdscache  dcecp  link
cdscp  directory  object
clearinghouse create

Creates a new clearinghouse on the local machine.

Format

clearinghouse create clearinghouse_name_list

Usage

The clearinghouse create operation creates a new clearinghouse on the local machine. The clearinghouse_name_list argument is a list of one or more names of the clearinghouses you want to create. Clearinghouses should only be named in the root directory — that is, /.:. This operation also stores a read-only replica of the root directory in the new clearinghouse. The process that creates the new clearinghouse starts a skulk of the root directory, so all replicas of the root must be reachable when you enter the clearinghouse create command. To ensure this, perform an immediate skulk of /.: prior to starting the command, using the directory synchronize /.: command. The command returns an empty string on success.

Privilege Required: You need w (write) permission to the server on which you intend to create the clearinghouse and a (admin) permission to the cell root directory. The server principal, /.:/hosts/hostname/cds-server, needs r (read), w (write), and a (admin) permission to the cell root directory.

Examples

The following command creates a clearinghouse named /.:/Boston_CH on the local host:

dcecp> clearinghouse create /.:/Boston_CH
dcecp>

Related Information

Commands:
cds cache dcecp link
cds cp directory object
clearinghouse delete

Deletes the specified clearinghouse from the local machine.

Format

clearinghouse delete clearinghouse_name_list

Usage

The clearinghouse delete operation deletes the specified clearinghouse from the local machine. The clearinghouse_name_list argument is a list of one or more names of the clearinghouses you want to delete. Clearinghouses that contain master replicas of directories are not deleted (and also return errors). This command also automatically deletes all read-only replicas from the clearinghouse; however, you should delete all read-only replicas by hand (see the directory delete -replica command) before starting this command because starting many skulks will cause the command to run more slowly. The command returns an empty string on success.

CDS does not permit you to delete a disabled (cleared) clearinghouse. Before you can delete a disabled (cleared) clearinghouse, you must recreate it using the clearinghouse create command.

Privilege Required: You must have w (write) and d (delete) permission to the clearinghouse and a (admin) permission to all directories that store replicas in the clearinghouse. The server principal, .:/hosts/hostname/cds-server, must have d (delete) permission to the associated clearinghouse object entry and a (admin) permission to all directories that store replicas in the clearinghouse.

Examples

The following command deletes a clearinghouse named .:/Orion_CH from the local host:

dcecp> clearinghouse delete .:/Orion_CH

dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>dcecp</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>directory</td>
<td>object</td>
</tr>
</tbody>
</table>
clearinghouse disable

Removes knowledge of the specified clearinghouse from the local server.

Format

clearinghouse disable clearinghouse_name_list

Usage

The clearinghouse disable operation removes knowledge of the specified clearinghouse from the local server. The clearinghouse_name_list argument is a list of names of one or more clearinghouses you want to disable. Use this command when relocating a clearinghouse. This command removes the name of the prefix of the clearinghouse files from the /opt/dcelocal/var/directory/cds/cds_files file and notifies the local CDS server that the clearinghouse is disabled. The clearinghouse entry is not removed from the name space, nor are the data files associated with the clearinghouse removed. The operation returns an empty string on success.

Privilege Required: You must have w (write) permission to the CDS server on which the clearinghouse resides.

Examples

The following command disables the clearinghouse /.:/Paris2_CH so that it can be moved to another server:

dcecp> clearinghouse disable /.:/Paris2_CH

dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>dcecp</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>directory</td>
<td>object</td>
</tr>
</tbody>
</table>
clearinghouse help

Returns help information about the clearinghouse object and its operations.

Format

clearinghouse help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each clearinghouse operation.
-verbose Displays information about the clearinghouse object.

Usage

The clearinghouse help command is used without an argument or option to returns brief information about each clearinghouse operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the clearinghouse object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the clearinghouse help command.

Examples

dcecp> clearinghouse help
catalog Returns the names of all clearinghouses in a cell.
create Creates the named clearinghouse.
delete Deletes the named clearinghouse.
disable Disables the named clearinghouse.
ninitiate Begins an action on the named CDS clearinghouse.
repair Repairs a component of the named CDS clearinghouse.
show Returns the attributes of a clearinghouse.
verify Verifies the consistency of the clearinghouse.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>
clearinghouse initiate

Begins a defined action on the specified clearinghouse on the local machine.

**Format**

```
clearinghouse initiate clearinghouse_name_list -checkpoint
```

**Options**

- **-checkpoint** Forces the clearinghouse to checkpoint to disk.

**Usage**

The `clearinghouse initiate` operation begins a defined action on the specified clearinghouse. The required `clearinghouse_name_list` argument is a list of one or more names of clearinghouses you want to start actions on. Only a checkpoint action is available. This operation returns an empty string on success.

**Privilege Required:** You need `w` (write) permission on the clearinghouse server and `a` (admin) permission on the cell root directory. The server principal, `/.:/hosts/hostname/cds-server`, needs `r` (read), `w` (write) and `a` (admin) permission on the cell root directory.

**Examples**

The following command begins a checkpoint operation on the clearinghouse named `/.:/oddball_ch` on the local system:

```
dcecp> clearinghouse initiate /.:/oddball_ch -checkpoint
dcecp>
```

**Related Information**

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>dcecp</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>directory</td>
<td>object</td>
</tr>
</tbody>
</table>
clearinghouse operations

Returns a list of the operations supported by the clearinghouse object.

Format

clearinghouse operations

Usage

The clearinghouse operations command uses no arguments, and returns a list of the available operations for the clearinghouse object. The order of the elements is alphabetic, except help and operations are listed last.

Privilege Required: No special privileges are needed to use the clearinghouse operations command.

Examples

dcecp> clearinghouse operations
catalog show help operations
dcecp>

Related Information

Commands:

- cdscache
- dcecp
- cdscp
- directory
- link
- object
clearinghouse repair

Repairs a specific problem on a specified clearinghouse on the local machine.

Format

clearinghouse repair clearinghouse_name_list -timestamps

Options

-timestamps Analyzes and repars incorrect timestamps found in a clearinghouse.

Usage

Use the clearinghouse repair operation to correct various problems that can occur in a clearinghouse. The required clearinghouse_name_list argument is a list of one or more names of clearinghouses you want to start repair actions on. Only timestamps that are not valid can be repaired. This operation returns an empty string on success.

Privilege Required: You need w (write) permission on the clearinghouse server and a (admin) permission on the cell root directory. The server principal, /:/hosts/hostname/cds-server, needs r (read), w (write), and a (admin) permission on the cell root directory.

Examples

The following command repairs incorrect timestamps in the clearinghouse named /:/blech_ch on the local system:

dcecp> clearinghouse repair /:/blech_ch -timestamps
dcecp>

Related Information

Commands:
cdscache       dcecp       link
cdscp           directory    object
clearinghouse show

Displays attribute and counter information associated with specified clearinghouses.

Format

clearinghouse show clearinghouse_name_list [-counters] [-attribute] [-all] [-schema]

Options

-all Returns the attributes and counters for the clearinghouse.
-attribute Returns the attributes for the clearinghouse.
-counters Returns the counters for the clearinghouse.
-schema Returns the name and type for each clearinghouse attribute.

Usage

The clearinghouse show operation displays attribute and counter information associated with specified clearinghouses. The required clearinghouse_name_list argument is a list of one more names of the clearinghouses you want to show. If more than one clearinghouse is shown, the attributes of all the clearinghouses are concatenated into one list. The order of the returned attributes is the alphabetic order of the object identifiers (OIDs) of each attribute for each clearinghouse.

When used without any options, dcecp clearinghouse show returns just the attributes associated with the specified clearinghouse. View just the clearinghouse counters by using the -counters option, or just the clearinghouse attributes by using the -attribute option. View attributes and counters by using the -all option. View just the name and type for each attribute by using the -schema option.

When using both -schema and either the -counters or -all options, the results from -counters are displayed along with -schema results. When using both -schema and the -attribute options, only the -schema results are displayed.

Privilege Required: You must have r (read) permission to the clearinghouse.

Examples

The following command displays the attributes of the clearinghouse l:/drkstr_ch.
The following command displays the current values of the counters associated with the /./Chicago1_CH clearinghouse:
dcecp> clearinghouse show /./Chicago1_CH -counters
corruptions 0
disables 0
enables 1
failedupgrades 0
missingentries 0
reads 2336
returnedrefs 2
rootunreachables 0
skulkfailures 0
writes 68
dcecp>

Related Information

Commands:
cdscache
dcecp
dcp
cds
link
directory
object
clearinghouse verify

Verifies the consistency of the specified clearinghouse on the local machine.

Format

```
clearinghouse verify clearinghouse_name_list
```

Usage

The `clearinghouse verify` operation verifies the consistency of the specified clearinghouse by checking internal attributes. The required `clearinghouse_name_list` argument is a list of one or more names of clearinghouses you want to verify. This operation returns an empty string on success.

**Privilege Required:** You need w (write) permission on the clearinghouse server and a (admin) permission on the cell root directory. The server principal, `/hosts/hostname/cds-server`, needs r (read), w (write), and a (admin) permission on the cell root directory.

Examples

The following command verifies the consistency of clearinghouses named `/./gumby_ch` and `/./pokey_ch`:

```
dcecp> clearinghouse verify {/.:/gumby_ch /./pokey_ch}
dcecp>
```

Related Information

Commands:

```
cdscache       dcecp       link
cdscp         directory      object
```
clock

A **dcecp** object that manages the clock on a local or remote host.

**Format**

```
clock compare [dts_entity] [-server dts_entity]
clock help [operation | -verbose] [-syntax]
clock operations

clock set [dts_entity] {-to DTS_timestamp [-abruptly -epoch number] | -epoch number}
clock show [dts_entity]
clock synchronize [dts_entity] [-abruptly]
```

**Parameters**

- **dts_entity**: The *clock* operations take an optional argument that can be the name or the binding of one *dtsd* server to act on. Usually, this represents a DTS entity on a remote host. With the *-server* option, *dts_entity* represents a DTS time provider. The name syntax is:

```
//.../cellname/hosts/hostname/dts-entity
```

The *dtsd* can also be specified with a string binding for the remote host on which the *dtsd* is running. Use a string binding such as

```
{ncacn_ip_tcp:130.105.1.227[4345]}
```

Or, you can specify the binding using **dcecp** string syntax such as:

```
{ncacn_ip_tcp 130.105.1.227 4345}
```

- **operation**: The name of one specific *clock* operation (subcommand) about which you want help information.

**Usage**

The *clock* object represents the clock on a system and the time that it tells. This object has commands to display and set the time. The time setting commands are done through the Distributed Time Service (DTS). The optional argument to these commands is the name of an OSF DCE Version 1.1 *dtsd* running on some machine. Without an argument, the _s(dts) convenience variable is checked. If not set, the commands operate on the clock on the local machine.

**Related Information**

Commands:

```
dcecp       dts         dtscp      utc
```
clock compare

Returns the difference between the clocks on the local machine and a time provider in the cell.

Format

clock compare [dts_entity] [-server dts_entity]

Options

-server dts_entity

Optionally names a specific time provider to compare the host clock against.

Usage

The clock compare operation returns the difference between the clocks on the local machine and a time provider in the cell. If a time provider is not running, the command picks the last responding server returned by dts catalog. An optional argument compares a remote host clock against a time provider. An optional -server option compares the clock against a specific time provider.

Privilege Required: You must have r (read) permission on the ACLs associated with the pertinent DTS entities: /.../cellname/hosts/hostname/dts-entity

Examples

dcecp> clock compare
{server /.../mycell.goodco.com/hosts/medusa/dts-entity}
{provider no}
{skew -0-00:00:00.256I-----}
dcecp>

Related Information

Commands:

dcecp dts dtscp utc
clock help

clock help

Returns help information about the clock object and its operations.

Format

clock help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each clock operation.
-verbose Displays information about the clock object.

Usage

The clock help command is used without an argument or option to return brief information about each clock operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the clock object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the clock help command.

Examples

dcecp> clock help compare
Returns the difference between the local clock and a server.
set
Sets the system clock to the specified time.
show
Returns the current time as a DTS style timestamp.
synchronize
Synchronizes the local dtsd with DTS servers.
help
Prints a summary of command-line options.
operations
Returns the valid operations for command.

Related Information

Commands:

dcecp dts dtscp utc
clock operations

Returns a list of the operations supported by the clock object.

Format

clock operations

Usage

The clock operations command uses no arguments, and returns a list of the available operations for the clock object. The order of the elements is alphabetic, except help and operations are listed last.

Privilege Required: No special privileges are needed to use the clock operations command.

Examples

dcecp> clock operations
compare set show synchronize help operations
dcecp>

Related Information

Commands:

dcecp dts dtscp utc
clock set

Sets the clock to the specified time or changes the DTS server epoch number.

Format

clock set [dts_entity] [-to DTS_timestamp [-abruptly -epoch number] | -epoch number]

Options

-abruptly Specifies to set the clock abruptly rather than gradually adjust it to the computed time. This option requires the -to and -epoch options.

-epoch number Specifies a number that matches the epochs of servers with which the local clock synchronizes. This option requires the -abruptly option if specified with the -to option.

-to DTS_timestamp Specifies a DTS timestamp as the time to set the clock. You can specify the time using the ISO compliant time format of CCYY-MM-DD-hh:mm:ss.fff

Usage

The clock set operation sets the local clock to the specified time, or changes the DTS server epoch number. An optional argument specifies to set the clock on a remote host. The -to option specifies a DTS timestamp as the time to set the clock. The clock is adjusted gradually by DTS to the specified time. The -abruptly option specifies that the time is immediately changed to the specified time. When this option is present, the -epoch option must also be specified to indicate a new epoch. The -epoch option with no other option only changes the epoch of the DTS server. An empty string is returned upon successful completion. Note that setting the system clock is a risky operation. The system clock must be synchronized with others in the cell for DCE services, such as CDS, to operate correctly.

Privilege Required: You must have w (write) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity

Examples

dcecp>

Related Information

Commands:

dcecp dts dtscp utc
clock show

Returns a DTS style timestamp including the Time Differential Factor.

Format

clock show [dts_entity]

Usage

The clock show operation returns a DTS style timestamp with the TDF specified. An optional argument specifies to show the clock on a remote host.

Privilege Required: You must have r (read) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity

Examples

dcecp> clock show
1996-07-15-16:28:02.229+00:00I-----
dcecp>

Related Information

Commands:
dcecp dts dtscp utc
clock synchronize

Causes the dtzd to synchronize gradually with time-providers.

Format

clock synchronize [dts_entity] [-abruptly]

Options

-abruptly Specifies to synchronize the clock abruptly rather than to adjust it gradually to the computed time.

Usage

The clock synchronize operation causes the local dtzd to gradually synchronize the local clock with the time from DTS servers. The machine clock is adjusted accordingly. An optional argument specifies to synchronize the clock on a remote host. An empty string is returned upon successful completion.

Privilege Required:  You must have w (write) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity

Examples

dcecp> clock synchronize
dcecp>

Related Information

Commands:

dcecp  dts  dtscp  utc
**dcecp**

An administrative interface for performing DCE management tasks. The interface accepts interactive commands and scripts written with the **dcecp** language.

**Format**

```
  dcecp [-s]
  dcecp [-s] script_name [arguments]
  dcecp [-s] -c command
```

**Parameters**

- **arguments**: Values passed to the user-created script. **dcecp** creates several script variables to enable the user-created script to access these values:
  - **argc**: The number of arguments.
  - **argv**: A list whose elements are the arguments.
  - **argv0**: The file name of the user-created script.

- **-c command**: The **command** is a valid **dcecp** command to be started in command line mode. See "Administration Objects" on page 122 for a description of the **dcecp** command format.

- **-s**: The **-s** option turns off inheritance of the login context, which is the default.

- **script_name**: The **script_name** is the file name of a user-created script containing **dcecp** commands.

**Usage**

The DCE control program is the primary DCE administration interface, providing remote access to routine DCE administrative functions from z/OS DCE.

**dcecp** provides consistent and uniform access to DCE administrative functions, wherever they reside, from any and every point in a cell. It performs all routine DCE operations from within a single administrative interface, managing most aspects of the DCE core components consistently and predictably, regardless of the platform on which they are run. **dcecp** implements most of the operations previously performed by using various component control programs, streamlining administration by providing a number of task objects for performing complex DCE operations.

The DCE control program is built on a portable command language called the Tool Command Language (Tcl). Tcl allows the use of variables, if statements, list processing functions, loop functions and many other features commonly found in command languages. The control program extends these features by providing a set of commands for manipulating specific DCE objects. The control program also includes task scripts to help administrators perform some routine DCE management functions. Refer to the [z/OS DCE Administration Guide](https://www.ibm.com/support/docview.wss?uid=swg21419899) for information about the basic concepts and features of **dcecp**. All of Tcl is included in the **dcecp** language.

Tcl commands are not described in this document. For further information, refer to your Tcl reference documentation (see the "Bibliography" on page 715).
Starting and Stopping DCECP

The DCE control program allows you to start dcecp commands in the following modes:

- Interactive mode
- Command line mode

Interactive Mode

Activate interactive mode by entering the dcecp command without any arguments. At the dcecp prompt, enter a dcecp or Tcl command. The dcecp program processes the command, displays the result, and is ready to accept another command. The following example starts a dcecp directory list command in interactive mode:

```
dcecp
```
```
dcecp> directory list /.: -directories
/.../mycell.goodco.com/hosts /.../mycell.goodco.com/subsys
```
```
dcecp>
```
End an interactive dcecp session using the exit or quit commands using the following command syntax:

- exit n
- quit

Use the *n* argument to specify the exit value returned to the shell. If no value is specified, exit passes the return value of the most recent command to the shell. The following example ends a session and returns an exit value of 56 to the shell:

```
exit 56
```

Command Line Mode

Activate command line mode from the system prompt using one of the following methods:

- Enter the dcecp command with a file name of a script containing dcecp commands (and other valid Tcl commands). You can use any extension or none at all. For example:

```
dcecp myown.tcl
```

Note that all script files must be in code page IBM-1047.

- Enter the dcecp command with the -c option followed by a dcecp command. DCE runs the command and then exits.

```
dcecp -c directory create /.:/admin/printers
```

Enter multiple dcecp commands by separating them with a semi-colon (;) and enclosing the commands in double quotations (""). Remember to escape shell metacharacters (for example by enclosing them in double quotations). Multiple commands must be on a single line as follows:

```
dcecp -c "directory create /.:/admin/printers;directory show /.:/admin/printers"
```

**Note:** When entering multiple commands, the results of each command is held internally within dcecp and only the results of the last command are returned to the shell. For example:

```
dcecp -c "name help;name get ncacn_ip_tcp:129.35.20.1"
```
```
hosts/medusa
```

To display the results of the other commands, use the Tcl puts command:
**dcecp**

-dcecp -c "puts [name help]; name get ncacn_ip_tcp:129.35.2wzerodot.1"

compare Compares two names syntactically.
expand Returns the canonical form of a name.
get Gets host name from a partial or full string binding
parse Parses name into cell name and residual name.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
hosts/medusa

By default, **dcecp** returns 0 (zero) on success and 1 if a command fails.

**dcecp** can be started in either interactive mode or command line mode from a batch program. See the section on running administration and user commands in batch in the [z/OS DCE Administration Guide](url) for more information. The **dcecp** subcommands and arguments entered in interactive mode and command line mode must be in the local code page. This also applies when running the **dcecp** command in batch.

## Startup Scripts

The following script files are sourced (run) during **dcecp** initialization, in the order shown.

**Note:** These (and all) **dcecp** script files must be in code page IBM-1047.

1. **init.tcl.**
   - By default in directory `/opt/dcelocal/tcl/`, but you can change the directory using the TCL_LIBRARY environment variable
   - Contains functions used to find non-**dcecp** commands

2. **init.dcecp**
   - In directory `/opt/dcelocal/dcecp/`
   - As shipped, sets `/opt/dcelocal_path` variable to the directory where several **dcecp** commands create and read files.
   - Can be used to customize **dcecp** for all users

3. **attr_eval.tcl**
   - In directory `/opt/dcelocal/dcecp/`
   - Contains function for obtaining an attribute value from a list

4. **dir_ops.dcecp**
   - By default in directory `/opt/dcelocal/dcecp/`, but can change directory by changing the `/opt/dcelocal_library` global variable; for example, in init.dcecp
   - Contains functions for handling directory trees

5. **.dcecp**
   - In the home directory of the user as defined by the HOME environment variable.
   - Used only when **dcecp** is started for interactive use, even when the commands are in a batch job.
   - Not shipped with DCE. Can be created for each user who needs specific **dcecp** customization

## Environment Variables

The env array is initialized to the current environment variables when **dcecp** is started. For example, env(HOME) is the value of the HOME environment variable. These values can be used and changed in **dcecp** scripts and interactive mode. Any change to an environment variable through its value in the env array is only known during the **dcecp** session; the original environment values are restored when **dcecp** ends.

For information on how to set environment variables, see the chapter on environment variables in the [z/OS DCE Administration Guide](url) SC24-5904.
There are several environment variables that `dcecp` uses in processing:

**DCECP_ERROR** Specifies whether additional information displays when an error is detected. ON causes the contents of the `errorInfo` and `errorCode` variables to display, if set, in addition to the usual error messages or output. Only the usual error messages or output display if another value or no value is specified for DCECP_ERROR.

**HOME** Specifies the home directory, needed to determine the location of any files specified with the tilde (`~`) expansion. In particular, when `dcecp` is started in interactive mode, HOME locates the `.dcecprc` initialization script file; an error message is issued if HOME is not set.

**PATH** Specifies the directories to check for an executable for a command that is not a `dcecp` or Tcl command. It is also used for resolving a command in which the object is abbreviated. An error message is issued if PATH is not set.

**TCL_LIBRARY** Specifies the directories to search for certain Tcl script files including `init.tcl`. If specified, this becomes the value returned by the `info library` command.

## Administration Objects

A `dcecp` command has the following syntax:

```
object operation [argument] [ -option [value]] . . .
```

where:

**object** Specifies the name of a `dcecp` administration object. Examples of administration objects are CDS directories, access control lists (ACLs), DTS servers, and server control objects. Each administration object is briefly in Table 1 on page 11.

**operation** Specifies the name of an action such as `create`, `show`, or `remove`, that is being performed on an administration object. For complete descriptions of operations supported by each `dcecp` object, refer to individual object reference pages. Common operations are briefly described in “Common Operations.”

**argument** Specifies the name of one or more specific objects to operate on. Most but not all `dcecp` objects take an argument. Refer to the individual reference pages for descriptions of the arguments supported by various objects.

**-option** Specifies a qualifier that controls the precise behavior of a `dcecp` command. Most but not all `dcecp` commands take options. Specify options by preceding the option name with a hyphen, as in `-replica`.

**value** Some options take an argument that can be a name or a value. The following example shows a `-clearinghouse` option and its argument, which is the name of a CDS clearinghouse. For example,

```
directory create /.:admin -clearinghouse /.:boston_ch
```

See “Syntax” on page 130 for information on using an alternate syntax.

## Common Operations

This section gives a description of operations as commands that are common to more than one object. Some operations presented here are implemented in all objects, some in only a few, and some only for specific types of objects; for instance, containers in CDS directories.

The descriptions in the sections on individual objects may override some of the information presented
here. Usually this is only in the form of an operation accepting more options, but other changes are possible.

**add**

Adds a member to a container. It is implemented for all objects that represent containers. Returns an empty string on success. The argument is a list of names of containers. The required `-member` option specifies the name of the member being added to the containers. Its value is a list of members being added. If lists are specified for both the `-member` option and as the argument, then each member name is added to each container. For example, it adds a member to an RPC group, and adds an element to an RPC profile.

**catalog**

Returns the names of all instances of an object. It usually uses no argument. In some cases, an argument specifying a scope, such as a cell name, is optional. For example, the `principal catalog` command returns a list of all principals in the registry. This is only implemented by those objects for which this is possible. By default, full names are returned. Some objects support a `-simplename` option that returns names in a shorter form (either relative or not fully qualified). The order of the returned list depends on the object.

**create**

Creates a new instance of an object. This command uses one argument by which a list of names of instances can be created. An empty string is returned upon successful completion. An error is returned if the object already exists. For some objects, this command uses an `-attribute` option or a set of attribute options to create attributes on the new object.

**delete**

Destroys an instance of the object. This command uses one argument by which a list of names of instances can be deleted. An empty string is returned upon successful completion. An error is returned if the object does not exist.

**help**

Returns help information on the object as described in the Help System section. This command uses an argument that may be an operation supported by the object. The `-verbose` option returns more information, and the `-syntax` option returns syntax diagrams.

**list**

Returns a list of the names of all the members of a container. This operation returns names of members, never any other information about the members; for example, attributes. It is implemented on all objects that represent containers. The argument is a list of names of containers, whose members are to be listed. The order of the returned list depends on the object. If more than one container name is given as an argument, all the member names are returned in one list.

**modify**

This operation is used to modify attributes, policies, counters, or any other information in an object. This means that all attributes, policies, counters, and so on, in any one object must have unique names. It is not available to all objects. An empty string is returned upon successful completion. The argument is a list of names of objects to modify. All objects are modified in the same way.

The specific modification is described by the use of one or more of the `-add`, `-remove`, or `-change` options. Some `modify` commands allow only one option to be specified; others allow all of them. If more than one is used, then the whole `modify` operation is treated atomically in that either it all works, or none of it does. Each option may only be used once for each command. If more than one attribute is being modified, then the value of that option should be an attribute list.

- **-add**
  Used to add an attribute to an object or to add values to an existing attribute. The value of this option is an attribute list.

- **-remove**
  Used to remove an entire attribute or some of its values from an object. The value of this option is an attribute list.
-change

Used to change one attribute value to another. The value of this option is an attribute list.

operations

Returns a list of the operations supported by the object. It uses no arguments, and always returns a Tcl list suitable for use in a foreach statement. The order of the elements is alphabetic, except help and operations are listed last. If you want them sorted, use the lsort Tcl command:

lsort [object operations]

remove

Removes an object from a container. It is implemented for all objects that represent containers. The argument is a list of names of containers. The remove command requires the -member option, used to specify the name of the member being removed from the container. The value is a list of names of members of the containers. If the value of this option and the argument to the command are both lists, then each listed member is removed from each specified container. An error message is returned if a member does not exist.

rename

This operation changes the name of a specified object. The argument is a single name of an object being renamed; it cannot be a list. This command uses a required -to option with a value of the new name. The value cannot be a list. An empty string is returned upon successful completion.

show

Returns information about an object instance. Objects can have various types of information such as attributes, counters, or policies. The show command returns any of this information. Options are passed to the command to specify what information is being returned. This command uses one argument, which is a list of names of instances being shown. Unlike the list command, which returns information about the members of a container, the show command only looks at the named object instance. If it is a container, it does not return information about the members, only the container itself.

synchronize

Tells the instance of an object to synchronize with any replicas of itself. In CDS terminology, this performs a skulk on a directory, in DTS it causes a server to synchronize. It is implemented for all objects that support replication. An empty string is returned upon successful completion. The argument is a list of instance names to synchronize. If more than one instance name is given, each instance synchronizes with all of its replicas. There is no relationship such as synchronize with each other, as this does not make sense for many objects.

Miscellaneous dcecp Commands

In addition to the object commands and the Tcl commands, the dcecp program includes a set of commands that you can use for performing miscellaneous operations.

echo

Evaluates and displays the supplied arguments. The syntax is:

echo [arguments]

For example:

dcecp> set a "/.:/hosts"
/.:/hosts
dcecp> echo directory list $a
directory list /.:/hosts
errtext

This command uses a DCE status code as an argument and returns the text of the associated message found in the message catalogs. The argument can be in decimal, octal (leading 0), or hexadecimal (leading 0x) notation. The syntax is:

```
errtext msgnum
```

msgnum is the status code or message number in decimal, octal (leading 0), or hex (leading 0x).

**Note:** Any substitution variables will display unknown contents. If you are displaying the message for a status code, there will be no leading message identifier.

For example:

```
dcecp> errtext 0x1131f003
EUVA04099E Cannot allocate storage.
```

login

Creates a new login context to be used for the rest of the dcecp session. This does not change the login context outside of dcecp. login sets convenience variables _c and _u to the cell name and principal name of the principal that issued the login command. Convenience variables are discussed separately within this section. One dcecp login context is kept. A subsequent login replaces any existing context created by dcecp login.

**Note:** Do not use dce_login (or dcelogin) to create a new login context from within dcecp. This will change the login context outside dcecp but not the login context within dcecp, and many dcecp commands will fail.

The klist command displays information for the original login context, whether that context is inherited by dcecp or not. It does not display information for any login context created by dcecp login.

**Note:** You cannot enter this command from the operating system prompt using dcecp -c login if you specify the -password option.

login requires an account name as an argument. It prompts for a password, which is not echoed to the screen, unless it is specified using the -password option. The syntax is:

```
login account_name [-password pw] [-certify]
```

- **account_name** Specifies the account whose login context will be logged in.
- **-password** Specifies the password of that account. You cannot specify the -password option if you are starting dcecp -c login from the operating system prompt. If -password is not specified, you are prompted for the account password.
  
  **Note:**

  If the password contains one or more blanks, do not specify the password; wait for the prompt.

- **-certify** Requests dcecp to certify the login granting source.

For example:

```
dcecp> login admin_user
Enter Password: <enter password>
dcecp>
```
logout
Logs you out of the current login context as established with a previous login command. Only the context created with a dcecp login can be logged out. Trying to log out of an inherited context results in an error. Also, leaving dcecp does a logout for any context created in the session. The syntax is:

```
logout
```

For example:

```
dcecp> logout
dcecp>
```

quit
Exits from dcecp. quit is a synonym of the Tcl command exit. The syntax is:

```
quit
```

For example:

```
dcecp> quit
/home/g123456/>
```

resolve
This command uses a partial string binding and returns a fully bound string binding. resolve uses a required -interface option and an optional -object option to provide enough information for the mapping to occur. The syntax is:

```
resolve str_binding -interface ifspec [-object uuid]
```

- **str_binding**: The partial string binding.
- **-interface**: The interface specification. With only an interface specification, resolve returns all endpoints that were exported for the specified interface.
- **-object**: The object specification. With both an interface and an object specification, resolve returns only the endpoint containing the specified object uuid.

For example:

```
ncadg_ip_udp 9.13Rzerodot.2Rzerodot.13 135
ncadg_ip_udp 9.13Rzerodot.2Rzerodot.13 1218
ncadg_ip_udp 9.13Rzerodot.2Rzerodot.13 1253
```

shell
Creates a command shell for the user. When the command shell ends, control is returned to dcecp. An error exception is generated if the shell exits abnormally. If called with arguments, they are passed to the shell and processed and control is returned at completion. The syntax is:
shell [arguments]

If no arguments are specified, the return value of the shell command is:

- The empty string if last command in shell before exit was successful
- Nothing if last command in shell before exit was not successful

If arguments are specified, the return value of the shell command is:

- The results of processing the arguments in the shell if successfully completed
- Nothing if processing the arguments in the shell was not successful

For example:

dcecp> shell
$ls
  dcecp  envar  krb5ccname  test.dcp
defs  hostname  pvar.dcp
$exit

dcecp>

Tcl Notes

The Tcl (Tool Command Language) included within dcecp is based on the Tcl 7.3 level, as described in "Bibliography" on page 715. However, there are several things to note in the dcecp implementation.

1. The source command allows the passing of parameters to the sourced script file. The source command syntax is

   source fn ?arg ...?

   The arguments are made available to the sourced script file using the following global variables:

   argc  Number of arguments
   argv  List whose elements are the arguments
   argv0  Name of the sourced script file

   The global variables will be reset by the next source command. Therefore, if your script file sources another script file or starts a command that sources another script file, make local copies of the global variables first.

2. Do not confuse the Tcl usage of & in file redirectors with the OMVS usage. You must follow the Tcl usage when in dcecp.

   In Tcl:

   `<@ fileid` and `@ fileid`   Redirect from/to an open file
   `> & fn`   Redirect standard output and standard error to a file

   In OMVS:

   `<& number` and `& number`   Redirect from/to a file descriptor

3. Put a blank before each redirector to ensure proper parsing.

   For example, use:

   exec cat <infile >outfile

   rather than

   exec cat <infile >outfile
4. In dcecp script files, avoid putting braces, brackets, or parentheses in comments. Tcl may count them as delimiters, even though they are in comments, resulting in errors because of unbalanced delimiters. Also, all dcecp script files must be in code page IBM-1047.

Command Processing

The dcecp control program supports the Tcl built-in commands as well as its own commands. If a command name is unknown to dcecp, it is passed to the unknown command (defined in the init.tcl script file) and dcecp evaluates it using the following algorithm:

- If the command is found in a dcecp script file, dcecp processes the command (using the exec subcommand), but the results are not saved.
- If the command exists as an executable UNIX program in a directory specified in the PATH environment variable, dcecp processes the command using the exec subcommand. Therefore, you can start any UNIX command from the dcecp prompt; for example, ls -l. Because you do not leave dcecp, you do not lose any context you have established.
- If you started the command at the top level of dcecp and the command requests c-shell-like history substitution (such as, !!, \<number\> or ^old^new), dcecp emulates c-shell's history substitution.
- If you started the command at the top level of dcecp and the command is a unique abbreviation for another command, dcecp starts the command.

Note: Many non-dcecp commands can adversely affect the dcecp environment and should not be started from within dcecp. In particular, do not start dce_login (or dcelogin) to change the login context. Instead, use the dcecp built-in login command. (See "Miscellaneous dcecp Commands" on page 124 for more information.) Also, do not attempt to start dcecp again from within dcecp.

When processing commands for Cell Directory objects, like directory, object, link, cdscache, and clearinghouse, you can pass backslashes to CDS by using two backslashes in the dcecp command. DCECP converts the two backslashes to one backslash; CDS gets the backslash and uses it as needed. The following example creates an object that contains a blank:

dcecp> object create "/./PhoneBook/Peter\\ Piper"

CDS gets

"./PhoneBook/Peter\ Piper"

and the backslash, in this case, escapes the blank. The resulting object contains a blank.

If you are running from the command line with the -c option and not from the dcecp> prompt, you can also escape the backslash (\) and quotations (" ) characters. For example,
dcecp. -c object create "/./PhoneBook/Peter\\\\ Piper"

Running Operating System Commands from DCECP

Although the DCE control program is versatile, there are times when you may want to use operating system commands to accomplish some simple (or even not-so-simple) operations. The Tcl exec command provides a way for scripts to perform external commands by creating a sub-process in which the command processes. The following example uses the exec command to retrieve the local host name, which is then established as a hostname variable and, subsequently, used in the script.
The `exec` command usually returns the results of the operation performed in the sub-process. However, you can use UNIX-style redirection symbols (`<`, `<<`, and `>`) to redirect standard input or standard output. You can also use the vertical bar (`|`) symbol to pipe the output through filters such as `nroff`, `sort`, or `grep`.

When used alone, the `exec` command is synchronous, meaning that the external command completes before the script continues processing. But, when a sub-process takes a long time to complete, for instance when you synchronize directories in a CDS cell, you can use the `exec` command with an ampersand (`&`) to push a sub-process into the background. The following example uses the `exec` command to send previously collected output to a printer. This lets your script continue without having to wait for the print command to complete.

```
exec lpr output.log &
```

**Note:** You do not have to specify `exec` if the command name does not conflict with the name of a `dcecp` or Tcl command or procedure.

```
ls -a .dcep .profile env
dcecp>
```

Interactive operating system commands, such as `OEDIT`, might not run correctly when started from the `dcecp` command line because of redirection of input and output. Interactive commands can be issued after entering the shell (by issuing the `shell` command).

### Abbreviations

`dcecp` makes use of two mechanisms to allow all object names, operation names, and options to be abbreviated to the shortest unique string in interactive commands.

The first mechanism relies on the `unknown` command, previously described in "Command Processing" on page 128. This requires setting the `PATH` environmental variable.

The second mechanism used for abbreviations is built into the individual `dcecp` commands themselves. This allows the operation name to be abbreviated to the shortest unique operation supported for an object, and the options to be abbreviated to the shortest unique string representing an option supported by an object and operation.

For example, consider the `directory create` operation:

```
directory create /.:/admin/printers/ascii -replica -clearinghouse SFO_CH
```

In the abbreviated form, the same operation can be entered as:

```
dir cre /.:/admin/printers/ascii -r -c SFO_CH
```

Although abbreviating commands is a good way to save entering interactive commands, abbreviations are not recommended for use in scripts. New procedures in scripts can cause abbreviations to become ambiguous. Furthermore, abbreviations are not always portable. When scripts are moved to other...
machines, some definitions may be left behind so scripts will not work correctly. Always spell out complete names in scripts.

Syntax

The **dcecp** commands have a default word order which is *object operation*. This order facilitates adding new objects because new objects can be added along with their operations.

**dcecp** is configured to accept commands ordered as *operation object* in addition to the standard *object operation* syntax. Because the names of some **dcecp** operations conflict with existing Tcl commands, these operations were given a second unique name when used in the alternate *operation object* syntax:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Alternate Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>dlist</td>
</tr>
<tr>
<td>rename</td>
<td>drename</td>
</tr>
<tr>
<td>set</td>
<td>dset</td>
</tr>
</tbody>
</table>

For example, the equivalent to

**directory list ./:/hosts**

in the alternate *operation object* syntax is

**dlist directory ./:/hosts**

**Note:** See the [z/OS DCE Administration Guide](#) for instructions on how to disable the alternate *operation object* syntax for all users. If the alternate syntax is disabled, you can enable it by adding the following line to your `.dcecprc` script file in your HOME directory:

```bash
source /opt/dcelocal/dcecp/alternate_syntax_script_name
```

where

*alternate_syntax_script_name*

is the name of the syntax script file. The default name is **verb-object.dcp**.

Do not use the alternate operation object syntax in a script file, because this limits its portability.

Argument Lists

Many commands can act on multiple objects by specifying a list of objects as the argument of the command. When a list of objects is specified, the action of the command is performed iteratively on each object in the list.

For example:

**organization remove {dceorg devorg} -member ken**

removes principal **ken** from organizations **dceorg** and **devorg**.

If the command fails for an object, all changes successfully made to the objects before the failing object remain in effect but usually no additional objects are processed and an error is returned. For a query command, fix the problem, then enter the command again. For other commands, fix the problem, then enter the command again using the part of the argument list that starts with the object in error.
Attribute Lists

Many commands can specify attributes to operate on. For example, the `modify` command allows attributes to be changed and the `create` command often allows attributes being created along with the object. In all cases, you can use an attribute list to specify the attributes and their values. This makes passing information from one command to another very easy. For example, an ACL copy operation can be written as follows:

```bash
# copy acl name1 to acl name2
# no error checking
proc acl_copy {name1 name2} {
    acl replace $name2 -acl [acl show $name1]
}
```

Attribute Options

While attribute lists are useful for writing scripts, they are often not user-friendly. For those objects that have a fixed list of attributes (principal and dts, but not object), wherever an attribute list is allowed, options for each attribute that are named the same as the attribute are allowed, followed by their values. For example, the following are equivalent:

```bash
principal create melman -attribute {{quota 5} {uid 123}}
principal create melman -quota 5 -uid 123
```

Lists of Lists

The `dcep` control program interpreter relies on list structures as a way to parse command input and return command output. For instance, the `-remove` option in the following example uses a list to group the attribute and value parts of the option argument together. The example is a command that removes some ACLs from an object called `./:/foo`:

```bash
acl modify ./:/foo -remove {user melman}
```

The argument to the `-remove` option is an ACL Entry. The ACL entry happens to be a list where the first element describes the ACL type, in this case `user`, and the second is the key for which user, in this case `melman`. However, the `-remove` option may take a list of ACL entries, so the following is valid as well:

```bash
acl modify ./:/foo -remove {{user melman} {user salamone}}
```

Lists of one value that do not contain spaces do not need braces. The string syntax of an ACL entry allows the type and key to be separated by a colon (:`), so the following are valid:

```bash
acl modify ./:/foo -remove user:melman
acl modify ./:/foo -remove {user:melman user:salamone}
```

If there is only one ACL entry given, that is, the `-remove` option's value has only one element and that element does not contain spaces, then braces are not needed to delimit the list. The following are all valid, but all are examples with unnecessary braces:

```bash
acl modify ./:/foo -remove {{user melman}}
acl modify ./:/foo -remove {{{user melman}}}  
acl modify ./:/foo -remove {user:melman}
acl modify ./:/foo -remove {{user:melman} {user:salamone}}
```
Specifying the Binding Information

The binding_info parameter specifies the binding, which can be a string binding, a server entry name, or a list containing one or more string bindings or server entry names.

The following example shows a server entry name binding:

```
./:hosts/host_name/dce_entity_name
```

The following example shows a string binding in standard syntax:

```
ncadg_udp_ip:130.105.96.3[1234]
```

The following examples show a string binding in TCL syntax:

```
{ncacn_ip_upd:10.29.58.00[2001]}
{ncacn_ip_upd 10.29.58.00 2001}
ncacn_ip_upd:10.29.58.00\[2001\]
```

Convenience Variables

All dcecp commands set several variables, called convenience variables, during processing. The variables contain the name of the object operated on, the return value of the last command, the cell name of the last object operated on, and so on. You can substitute the value of these variables into the next command to save entering.

Convenience variables act just like other variables in dcecp. Therefore, you can trigger variable substitution by adding a dollar sign ($) before the name of the variable. You can also trigger substitution using set. Some convenience variables can only be set by the dcecp program. Users can set the others. See the [z/OS DCE Administration Guide] for more information.

The dcecp program defines the following variables:

- **_b**
  Holds the name of the server bound to for the last command. This is actually a Tcl array where the indexes identify the service. There is only one defined index: sec.
  The value specifies the name of a server in whatever manner the service finds useful. This can be the name of an RPC server entry in the name space, or a string binding, or the name of a cell. This variable may not be set by the user.

- **_c**
  Holds the cell name of the current principal. The login command sets the cell name (_c) and principal name (_u) convenience variables at login (see the login command). This variable may not be set by the user.
  **Note:**
  _c and _u are set when dcecp is started if you are already logged in to DCE and dce_login was done from where dcecp was started (in z/OS UNIX System Services shell or TSO/E).

- **_h**
  Holds the DCE name of the current host. It is set when dcecp is started and cannot be changed by users.

- **_n**
  Holds a list of the names entered to the last command. These names are the names that the command operated on, typically entered as the third argument. Users can set this variable. An example of using the _n variable:
_o  Holds the object used in the last operation. For example, if the last command was
dir show /.:
then _o is directory. This variable may not be set by the user.

_p  Holds the parent of _n. If _n is a list, then this is a list of the same length, where each
element is the parent of the corresponding element in _n. _p is set to the empty string
if _n has no parent. This variable may not be set by the user.

_r  Holds the return value of the last processed interactive command, such as a return
code, output, or an error message. This variable is only set when dcecp is started in
interactive mode. This variable may not be set by the user.

_s  Holds the name of the server to use for the next command. This is actually a Tcl array
where the indexes identify the service. The defined indexes are: sec, cds, dts, and
aud.

The value specifies the name of a server in whatever manner the service finds useful.
This can be the name of an RPC server entry in the name space, or a string binding, or
the name of a cell. The variable is not set by dcecp. Users can set this variable by
issuing the set command. This lets users select which server is used.

The values of this variable (array) are treated differently by each service. For example,
the security service uses this variable to display the registry bound to for the last
command, and is used as a default for the next registry operation. If bound to a
read-only replica and an update is requested, dcecp tries to bind to the master registry
to perform the change. The CDS service only attempts to communicate with the CDS
server named by the variable. If the named CDS server cannot satisfy a request for
any reason, the request fails. The auditing service and DTS use their variables in a
similar manner to the CDS server. To contact an audit daemon or DTS server on
another host, set this variable to identify that server.

For information about an object's use of this variable, see the object's reference page or
use the object's help -verbose operation.

_u  Holds the current principal name. The login command sets the cell name (_c) and
principal name (_u) convenience variables at log in (see the login command). This
variable may not be set by the user.

Note:
_c and _u are set when dcecp is started if you are already logged in to DCE and
dce_login was done from where dcecp was started (in z/OS UNIX System Services
shell or TSO/E).

Error Handling and Debugging

All commands in dcecp return either a list of some information or an empty string on success completion.
If an error occurs, dcecp returns an error message. Tcl also provides a catch command to help dcecp
scripts catch errors and start error handlers.

The dcecp program provides two global variables that store additional error information returned from
The `errorInfo` variable contains the stack trace of the error messages, providing the precise location of the error and a trace of what processing was taking place. The `errorCode` variable, if set, usually contains error information from a system function that was called as part of a command (such as `open`).

As with any `dcecp` variable, you can enter `set errorCode` and `set errorInfo` to display their current values. In addition, there are two ways to automatically display the values:

1. Set the DCECP_ERROR environmental variable to ON. This displays the values of `errorInfo` and `errorCode` (if set) in addition to the regular output whenever an error is found. This includes errors found by `dcecp` when started in both command line and interactive mode, and when sourcing script files. DCECP_ERROR can be set before starting `dcecp` or it can be set or unset during `dcecp` processing by changing the value of the env(DCECP_ERROR) variable.

2. Set the `dcecp_verbose_errors` variable to 1 for the value of `errorInfo` to display instead of the regular error output whenever an error is found. `dcecp_verbose_errors` cannot be set before `dcecp` is started. If you want it to be active when starting `dcecp` in command line mode, add it to the source file or command that is being processed.

**Help**

The `dcecp` program provides several kinds of help. All return help strings obtained from appropriate message catalogs.

To see which operations an object supports, enter an `object operations` command. All `dcecp` objects support the `operations` command. An example is:

```
dcecp> principal operations
```

This provides simple help similar to usage messages found on many systems. Users unsure of an operation name or if an operation is supported by an object can use this command to find the answer. The output is a `dcecp` list that can be used by other `dcecp` commands.

To see other information about an object, use an object's `help` command. All `dcecp` objects have a `help` command that offers four kinds of information.

- View brief information about an object's operations using `help` without arguments or options. Operations are listed in alphabetic order, with the `operations` and `help` commands listed last, because all objects support these commands. An example is:

```
dcecp> principal help
```

- View brief information about options supported by an operation using `help` with one argument — the name of the operation. This command returns attribute options followed by other options. If no options are supported, an informational message is returned. An example is:
dcecp> principal help create
  -alias       Add principal named as an alias of specified uid.
  -attribute   Attribute list to be assigned to the new principal.
  -fullname    Fullname of the new principal.
  -quota       Quota of the new principal.
  -uid         User Identifier of the new principal.
  -uuid        Orphaned UUID to be adopted by the specified principal.

dcecp>

- View a short description of a **dcecp** object using the `help` command with the `-verbose` option. This command returns text explaining what the object represents and how to use it. An example is:

dcecp> principal help -verb
This object allows remote manipulation of principal information stored in the DCE registry. The argument is a list of either relative or fully qualified principal names. Specify fixed attributes using attribute options or attribute lists. Specify any extended attributes using attribute lists. Principal operations connect to a registry that can service the request. Specify a particular registry by setting the _s(sec) convenience variable to be a cell-relative or global replica name, or the binding of the host where the replica exists. The completed operation sets _b(sec) to the name of the registry contacted. Access to most operations relies solely on the ACL of the principal. Create requires i permission to the principal directory. Delete requires rD permission to the principal as well as d permission to the principal directory. Catalog requires r permission to the principal directory. Modify requires rfmu permission to the principal. Rename requires rf permission to the principal. Show requires r permission to the principal.

dcecp>

- View the syntax diagram for each of the object's operations using the `help` command with the `-syntax` option. An example is:

dcecp> principal help -syntax
principal catalog [server_name_list] [-simplename]

principal create principal_name_list
  [-attribute attribute_list | attribute options]

principal delete principal_name_list

principal modify principal_name_list
  {[-change attribute_list | attribute options] |
   -add extended_registry_attribute_list |
   -remove extended_registry_attribute_list [-types]}

principal rename principal_name -to new_principal_name

principal show principal_name_list [-xattrs] [-all]

principal help [operation | -verbose] [-syntax]

principal operations
dcecp>

- View the syntax diagram for a specific operation by specifying both the **operation** and **-syntax** in the `help` command. An example is:
dcecp

\[\text{dcecp}\geq \text{principal rename -syntax}\]
principal rename principal_name -to new_principal_name
\[\text{dcecp}\geq\]

**Command History and Command Line Recall**

The **dcecp** program includes a history facility that stores previously entered commands during an interactive **dcecp** session. View the stored commands using the **history** command.

By default, the history facility stores the 20 most recent commands, but you can use a **history keep** command to change this as in:

\[\text{dcecp}\geq \text{history keep 50}\]

Each stored command is numbered so you can enter it again using an exclamation point (!) followed by the event number.

\[\text{dcecp}\geq \text{17}\]
OUTPUT FROM WHATEVER EVENT 7 WAS.
\[\text{dcecp}\geq\]

Enter a specific command again using an exclamation point (!) followed by the first unique characters of a previously entered command:

\[\text{dcecp}\geq \text{!dir}\]
OUTPUT FROM WHATEVER EVENT dir WAS.
\[\text{dcecp}\geq\]

You can also enter the most recent command again and revise it using the ^old^new syntax, familiar to UNIX users:

\[\text{dcecp}\geq \text{directory vreate ./:/admin/printers}\]
EUVAO4104E Unrecognized argument 'vreate'.
\[\text{dcecp}\geq \text{^vreate^create}\]
\[\text{[ COMMAND OUTPUT ]}\]
\[\text{dcecp}\geq\]

**Examples**

The following examples show some ways to enter **dcecp** commands.

- Start **dcecp** for interactive use:
  \[\text{dcecp}\]
  \[\text{dcecp}\geq\]

- Start **dcecp** for a single command:
  \[\text{dcecp}\geq \text{c clock show}\]
  1996-04-21-19:12:42.203+00:00I-----

- Start **dcecp** and run a script:
  \[\text{dcecp}\geq \text{get_users.Tcl}\]
Simple Object Commands

dcecp> acl show -ic /.:  
{unauthenticated r--t---}  
{group subsys/dce/cds-admin rwdtcia}  
{group subsys/dce/cds-server rwdtcia}  
{any_other r--t---} 
dcecp> 

dcecp -c directory show /.:/subsys 
{RPC_ClassVersion {01.00}}  
{CDS_CTS 1995-10-26-15:09:09.085929100/00-00-00-00-00-00-00}  
{CDS_UTS 1995-10-31-11:08:26.703886100/00-00-00-00-00-00-00}  
{CDS_ObjectUUID 3e480686-0fa8-11cf-a589-10005ab169d0}  
{CDS_RepliCasts  

{{CH_UUID 20d1d050-0fa8-11cf-a589-10005ab169d0}  
{CH_Name /.../mycell.goodco.com/mycell2_ch}  
{Replica_Type Master}  
{Tower {ncacn_ip_tcp 9.130.79.118}}}  
{Tower {ncadg_ip_udp 9.130.79.118}}}  
{CDS_AllUpTo 1995-11-01-11:08:26.571}  
{CDS_ParentPointer  

{{Parent_UUID 21fa1758-0fa8-11cf-a589-10005ab169d0}  
{Timeout  

{expiration 1995-11-01-11:08:26.571}  
{extension +1-00:00:00.00010.000}}  
{myname /.../mycell.goodco.com/subsys}}}}  
{CDS_DirectoryVersion 3.0}  
{CDS_RepliCasts on}  
{CDS_RepliCasts Type Master}  
{CDS_LastSkulk 1995-10-31-15:08:27.035094100/00-00-00-00-00-00-00}  
{CDS_LastUpdate 1995-10-31-11:08:26.703886100/00-00-00-00-00-00-00}  
{CDS_Epoch 3e52ee92-0fa8-11cf-a589-10005ab169d0}  
{CDS_RepliCastsVersion 3.0}  

The foreach Loop

dcecp> foreach i [group list temps] {  
> account modify $i -expdate 1999-06-30 }
Abbreviations

dcecp> clearin sh /.../brain_cell.osf.org/pmin17_ch
{CDS_CTS 1996-01-31-19:27:15.721447100/00-00-00-00-00-00-00-00}  
{CDS_UTS 1996-01-31-19:29:58.424322100/00-00-00-00-00-00-00-00}  
{CDS_ObjectUUID 211d560b-13ba-11cf-b02c-08005ac1d81d}  
{CDS_AllUpTo 1996-04-07-20:52:59.677122100/f0-37-c2-b0-2f-f2}  
{CDS_DirectoryVersion 3.0}  
{CDS_CHName /.../brain_cell.osf.org/pmin17_ch}  
{CDS_CHLastAddress  
  {Tower {ncacn_ip_tcp 9.130.44.126}}}  
  {Tower {ncadg_ip_udp 9.130.44.126}}}  
{CDS_CHState on}  
{CDS_CHDirectories  
  {{Dir_UUID 21be04fa-13ba-11cf-b02c-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org}}  
  {{Dir_UUID 374a4e8c-13ba-11cf-b02c-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/subsys}}  
  {{Dir_UUID 37899aec-13ba-11cf-b02c-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/subsys/HTP}}  
  {{Dir_UUID 37c2c99b-13ba-11cf-b02c-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/subsys/HTP/sample_apps}}  
  {{Dir_UUID 383731ca-13ba-11cf-b02c-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/hosts}}  
  {{Dir_UUID 386e34fe-13ba-11cf-b02c-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/hosts/pmin17}}  
  {{Dir_UUID 13bf968d-1477-11cf-9b3b-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/users}}  
  {{Dir_UUID 14674678-1477-11cf-9b3b-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/users/newuser}}  
  {{Dir_UUID bae130d8-1943-11cf-9b3b-08005ac1d81d}  
    {Dir_Name /.../brain_cell.osf.org/users/xyz}}}  
{CDS_ReplicaVersion 3.0}  
{CDS_NSCellname /.../brain_cell.osf.org}  

dcecp>
directory

A dcecp object that manages a name service directory.

Format

directory add directory_name_list -member child_pointer_list -clearinghouse clearinghouse_name

directory create directory_name_list [-attribute attribute_list [-single]]
[-clearinghouse clearinghouse_name [-replica]]

directory delete directory_name_list [-tree | -clearinghouse clearinghouse_name [-replica]]

directory help [operation | -verbose] [-syntax]

directory list directory_name_list [-directories] [-objects] [-links] [-simplename]

directory merge source_directory_name -into destination_directory_name
[-clearinghouse clearinghouse_name] [-tree] [-nocheck]

directory modify directory_name_list
{[-add attribute_list [-single]] [-remove attribute_list [-types]] [-change attribute_list]}

directory operations

directory remove directory_name_list -member child_pointer_list

directory show directory_name_list
[-member child_pointer_list | -clearinghouse clearinghouse_name [-replica]] [-schema]

directory synchronize directory_name_list

Parameters

directory_name_list
   A list of one or more specific directory names being operated on.

operation
   The name of one specific directory operation (subcommand) about which you want help information.

source_directory_name
   The name of one specific directory whose contents are being copied into a destination directory using a directory merge operation.

Usage

The directory object represents CDS directories. CDS directories are containers for other objects, links, and other directories (as well as clearinghouses). Any of these items that reside in a directory are called children of that directory. Directories also contain attributes that may be viewed or modified.

This object also represents CDS replicas. Replicas are read-only copies of directories stored in other clearinghouses. Several of the supported operations take options to specify that the command operates on a specific replica.

If the _s(cds) convenience variable is set, it is treated as the name of a clearinghouse to contact for this operation. This is the only clearinghouse contacted in an attempt to complete the operation. These commands do not set the value of this variable after completion. If a -clearinghouse option is used (as described in some commands below), then it overrides the value of _s(cds), but the command does not change the setting of _s(cds).
Attributes

The following are the CDS defined attributes that may be present in directories and replicas in CDS. Unless otherwise specified, values for these attributes are set by the system and cannot be modified by the user.

**CDS_AllUpTo**

Specifies the date and time of the last successful skulk on the directory. All replicas of the directory are guaranteed to receive all updates whose timestamps are less than the value of this attribute.

**CDS_Convergence**

Specifies the degree of consistency among replicas. This attribute value is defined as one of the following:

- **low**
  
  CDS does not immediately propagate an update. The next skulk distributes all updates that occurred since the previous skulk. Skulks occur at least once every 24 hours.

- **medium**
  
  CDS attempts to immediately propagate an update to all replicas. If the attempt fails, the next scheduled skulk makes the replicas consistent. Skulks occur at least once every 12 hours.

- **high**
  
  CDS attempts to immediately propagate an update to all replicas. If the attempt fails (for example, if one of the replicas is unavailable), a skulk is scheduled for within one hour. Skulks usually occur at least once every 12 hours. Use this setting temporarily and briefly, because it uses extensive system resources.

By default, every directory inherits the convergence setting of its parent at creation time. The default setting on the root directory is **medium**. This is a single-valued attribute, and can be modified.

**CDS_CTS**

Specifies the creation timestamp (CTS) of the CDS directory.

**CDS_DirectoryVersion**

Specifies the current version of the directory (derived from the **CDS_DirectoryVersion** attribute of the clearinghouse in which the directory was created). Multiple directory versions are supported in a cell.

**CDS_Epoch**

A UUID that identifies a particular instance of the directory.

**CDS_GDAPointers**

A set-valued attribute that is only present in the root directory of a cell. This attribute contains location information about registered Global Directory Agents for that cell, similar to the CDS_Repliecaps attribute. It is created and used only by a GDA. This is a multi-valued attribute, and can be modified.

**CDS_InCHName**

Specifies whether a directory or any of its descendants can store clearinghouse names. If this value is **true**, the directory can store clearinghouse names. If it is **false**, the directory cannot store clearinghouse names. This is a single-valued attribute, and can be modified.

**CDS_LastSkulk**

Records the timestamp of the last skulk performed on this directory.

**CDS_LastUpdate**

Records the timestamp of the most recent change to any attribute of a directory replica, or any change to an entry in the replica.

**CDS_ObjectUUID**

Specifies the unique identifier of the directory. This is read-only to the user and is set by the system at creation time.

**CDS_ParentPointer**

Contains a pointer to this directory's parent in the name space.
CDS_Replicas  Specifies the address, UUID, and name of every clearinghouse where a copy of this directory is located. This attribute also specifies whether the replica in a particular clearinghouse is a master or read-only replica.

CDS_ReplicaState  Specifies whether a directory replica can be accessed.

CDS_ReplicaType  Specifies whether a directory replica is a master or read-only replica.

CDS_ReplicaVersion  Specifies the version of a replica of the directory.

CDS_RingPointer  Specifies the UUID of a clearinghouse containing another replica of this directory.

CDS_UpgradeTo  A single-valued attribute used to control upgrading a directory from one version of CDS to another. By modifying this attribute, you can upgrade a directory to a newer version of CDS.

CDS_UTS  Specifies the timestamp of the most recent update to an attribute of the directory.

Related Information

Commands:

- cdscache
- clearinghouse
- link
- cdscp
- dcecp
- object
directory add

Creates a child pointer in the parent directory.

Format

directory add directory_name_list -member child_pointer_list -clearinghouse clearinghouse_name

Options

-clearinghouse clearinghouse_name
   This required option names the clearinghouse containing a replica of the child directory.

-member child_pointer_list
   This required option names one or more child pointers being added to parent directories.

Usage

A directory add creates a child pointer in the parent directory. This command is only needed to recreate a child pointer that was accidentally deleted, such as in an error checking situation. Usually, child pointers are created internally by CDS when creating directories with the directory create command.

The directory_name(s) argument is a list of names of one or more parent directories to add child pointers to. The value of the required -member option is a list of names of child pointers to add to each of the directories listed in the argument. Each child pointer name should contain only the last RDN of the name. The child object must exist or the command returns an error. The full name of a clearinghouse that holds a replica of the child directory is given as the value to the required -clearinghouse option. This option may only have one value and is used for each of the values of the -member option. An empty string is returned upon successful completion. An error is returned if a child pointer of the same name already exists.

Privilege Required: You must have i (insert) permission to the parent directory.

Examples

dcecp> directory add /.: -member foo -clearinghouse /.:/darkstr_ch

dcecp>

Related Information

Commands:

cdscache clearinghouse link
cdscp dcecp object
directory create

Creates a new directory of the specified name.

Format

directory create directory_name_list [-attribute attribute_list [-single]]
[-clearinghouse clearinghouse_name [-replica]]

Options

-attribute attribute_list
   Lets you specify the CDS_Convergence attribute or the CDS_UpgradeTo attribute in
   an attribute list. The format is:
   -attribute {{attr value} {attr value}}
   See "Attributes" on page 140 for a description of the CDS_Convergence attribute and
   the CDS_UpgradeTo attribute.

-clearinghouse clearinghouse_name
   Required with the -replica option and optional when the -replica option is not present,
   the -clearinghouse option names the clearinghouse where the child pointers are being
   added.

-replica
   This option specifies that the directory created is a replica of an existing directory. If
   you use the -replica option, you must specify a clearinghouse using the
   -clearinghouse option.

-single
   Valid only with the -attribute option, this option specifies that attribute values are
   single-valued. Otherwise, attributes are multi-valued.

Usage

directory create creates a new directory of the specified name. The directory_name_list argument is a
list of names of directories to be created.

An optional -attribute option specifies a list of attributes to include in each created directory. The attribute
values are multi-valued unless the -single option is specified, in which case all attributes are
single-valued. The -single option is valid only if the -attribute option is specified.

The -clearinghouse option specifies one clearinghouse in which to create all of the directories. The value
of this option is not a list. It is only one clearinghouse name to create all the directories named in the
argument. If this option is not specified, the new directories are created in the same clearinghouse as the
parent directory. The directory create command also uses a -replica option, which specifies that a
directory replica is created. When this option is used, the -clearinghouse option is required. An empty
string is returned upon successful completion.

Privilege Required: You must have the following permissions in order to create a directory:

- r (read) and i (insert) permissions to the parent directory
- w (write) permission to the clearinghouse in which the master replica of the new directory is being
  stored

In addition, the server principal, /./hosts/hostname/cds-server, must have r (read) and i (insert)
permission to the parent directory.
directory create

Examples

The following command creates a directory named /.:sales:

dcecp> directory create /.:sales
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>clearinghouse</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>dcecp</td>
<td>object</td>
</tr>
</tbody>
</table>
directory delete

Deletes a directory.

Format

directory delete directory_name_list [-tree | -clearinghouse clearinghouse_name [-replica]]

Options

- **-clearinghouse clearinghouse_name**
  Required with the -replica option, the -clearinghouse option names one clearinghouse (not a list of clearinghouses) in which the replica is being deleted.

- **-replica**
  Specifies that the directory being deleted is a replica of an existing directory. If you use the -replica option, you must specify a clearinghouse using the -clearinghouse option.

- **-tree**
  Removes the directory and everything beneath it (all directories, objects, links, and clearinghouses).

Usage

The directory delete operation deletes a directory from the CDS name service. The directory_name_list argument is a list of names of directories being deleted. If the directory is not empty, the command returns an error unless the -tree option is used. The -tree option, which uses no values, specifies to remove the directory and everything beneath it (all directories, objects, links, and clearinghouses).

To delete a replica instead of a directory, use both the -replica and -clearinghouse options. The clearinghouse the replica is in is specified as the value the -clearinghouse option; only one value can be specified, not a list. An empty string is returned upon successful completion. If a specified directory does not exist, an error is generated.

The -replica and -clearinghouse options cannot be used with the -tree option. If used alone, the -clearinghouse option is ignored.

Privilege Required: You must have d (delete) permission to the directory and w (write) permission to the clearinghouse that stores the master replica of the directory. The server principal, l:/hosts/hostname/cds-server, needs a (admin) permission to the parent directory or d (delete) permission to the child pointer that points to the directory you intend to delete.

Examples

The following command deletes the directory l:/eng from the name space:

dcecp> directory delete l:/eng

The following command tries to delete the non-empty directory l:/depts/phrenology, and receives an error. The second attempt uses the -tree option to delete the directory and all of the directories and objects beneath it.

dcecp> dir delete l:/depts/phrenology
EUVA07172E   DCE interface error.
   DCE status code: 0x10d0a3fc - Directory must be empty.
dcecp> dir delete l:/depts/phrenology -tree
dcecp>
Related Information

Commands:

- cdscache
- cdscp
- clearinghouse
- dcecp
- link
- object
directory help

Returns help information about the directory object and operations.

Format

directory help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each directory operation.
-verbose Displays information about the directory object.

Usage

The directory help command is used without an argument or option to return brief information about each directory operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the directory object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the directory help command.

Examples

dcecp> directory help
add Creates a child pointer in the specified directory.
create Creates the named directory.
delete Deletes the named directory.
list Lists the descendants of a directory.
merge Merges the contents of one directory into another.
modify Adds, removes, or changes an attribute in the named directory.
remove Removes a child pointer in the specified directory.
show Returns the attributes of a directory or replica.
synchronize Skulk the named directory.
help Prints summary of command-line options.
opinations Returns the valid operations for command.
dcecp>

Related Information

Commands:
cdsccache clearinghouse link
cdscp dcecp
cdscp object
directory list

Returns a list of the names of all the immediate descendants of a directory.

Format

directory list directory_name_list [-directories] [-objects] [-links] [-simplename]

Options

-directories Lists the names of all descendant directories.
-links Lists the names of all descendant softlinks.
-objects Lists the names of all descendant objects.
-simplename Specifies that the names are relative, not fully qualified.

Usage

The directory list operation returns a list of the names of all the immediate descendants of a directory. Descendants can include all directories, objects, links, and clearinghouses of the directory. The directory_name_list argument is a list of names of directories to be operated on. This command returns only the names of immediate descendents, so there is no way to tell the class of each name unless by convention; for instance, most clearinghouses end with _ch. Use the following options to specify the types of descendents to return:

-directories
-objects
-links

The options use no values and can be used in combination. By default, full names are returned. Use the -simplename option to return the name that is relative, not the fully qualified name.

Privilege Required:  You must have r (read) permission to the directory named in the argument.

Examples

The following example lists a softlink in the /.:depts/administration directory that points to a server entry in the /.:depts/phrenology/applications directory.

dcecp> dir list /.:depts/administration -link
/.../ward_cell.osf.org/depts/phrenology/applications

dcecp>

Related Information

Commands:

cdscache clearinghouse link
cdscp dcecp object
directory merge

Copies the contents of one directory into another directory.

Format

directory merge source_directory_name -into destination_directory_name
[-clearinghouse clearinghouse_name] [-tree] [-nocheck]

Options

-clearinghouse clearinghouse_name]
    Places the new objects (the resulting merged directory) in a clearinghouse other than
    that of the destination directory.

-into destination_directory_name
    Specifies the name of the destination directory; one specific directory whose contents
    are being copied from a source directory using a directory merge operation. The
    destination directory must exist.

-nocheck
    Lets the directory merge operation proceed without checking first for object name
    collisions or ACL problems. Use this option to save time when you are sure problems
    do not exist.

-tree
    Copies the contents of child directories as well as the child directories themselves into
    the destination directory.

Usage

The directory merge command copies the contents of one directory into another. The argument is the
name of the source directory. This command uses a required -into option to specify the destination
directory that must exist. For example, if /.:a has two child objects, /.:a/b and /.:a/c, then a directory
merge /.:a -into /.:x results in the following objects (assuming no errors):

/.:x/b and /.:x/c

Usually, only the immediate contents of the directory are merged. This means all objects, links, and
directories; but not the contents of child directories. To merge these as well, use the -tree option.

By default, the new objects are placed in the destination directory's clearinghouse, and all children (no
matter how many levels down) are placed in the same clearinghouse. To place any descendant
directories in another clearinghouse, use the -clearinghouse option with a value. There can only be one
clearinghouse specified for all directories involved in the merge operation. To specify more than one,
either change this after the merge has happened, or use separate commands.

This command first checks for any collisions or ACL problems before beginning to merge any objects. If
there are any problems encountered, an error is generated (not immediately, but after all objects are
checked) and the names of all problem objects, links or directories are returned in a list. The
administrator then addresses these problems and reruns the merge command. If the -nocheck option is
specified, the check is not performed. This way, you can save time when trying a merge with no known
problems. This is not an atomic operation, and other changes to the involved objects can cause
problems. Use this command when others are not modifying the involved directories. Changing ACLs
can be done to ensure this. If an error occurs during the actual merging process, it is generated and the
operation ends immediately.
directory merge

The merge command actually re-creates the objects with the same writable attributes of the source objects. This means that some read-only attributes change between the source and destination. For example, the creation timestamp attribute (CDS_CTS) changes.

The resulting merged directory inherits its ACLs from the destination directory's initial container or initial object ACLs, and the ACL of the destination objects may differ from the ACLs of the source objects.

Privilege Required: You must have r (read) permission to the source directory and r (read) and i (insert) permission to the destination directory.

Examples

The following command merges the directories, but not the contents of the /.:/depts/phrenology directory into the /.:/depts/radiology directory:

dcecp> dir list /.:/depts/phrenology -simple
applications services staff users

dcecp> directory merge /.:/depts/phrenology -into /.:/depts/radiology

dcecp> dir list /.:/depts/radiology -simple
applications services staff users

dcecp>

Related Information

Commands:

cdscache clearinghouse link
cdscp dcecp object
directory modify

Adds, removes, or changes directory attributes and their values.

Format

directory modify directory_name_list
{[-add attribute_list [-single]] [-remove attribute_list [-types]] [-change attribute_list]}

Options

-add attribute_list This option adds a value to a modifiable, set-valued attribute (including application-defined attributes) of a directory. If you enter a byte data type, you must enter an even number of digits in length. You can only enter pairs of hexadecimal values for user-defined attributes.

-change attribute_list
This option changes the value of a modifiable attribute of a directory.

-remove
This option removes a value from a set-valued or single-valued attribute of a directory (including application-defined attributes). If you do not specify a value, the command removes the entire attribute. This command can delete attributes created with the -add and -change options.

You cannot remove a single valued attribute like:
dcecp -c directory modify ./:/DirName -remove {dirregion 6}

You can only remove a multi-valued value; not the value of a single-valued attribute. However, if you want to remove the attribute altogether, use the -types option. This works for both multi and single valued attributes.

-single
When used with the -add option, specifies that the attributes being added are single-valued. Usually, all user defined attributes are defined as multi-valued, even if only one value is specified. This option is not valid without the -add option.

-types
When used with the -remove option, specifies that the value of the -remove option is a list of attribute types. This means that the entire attribute is removed, not just a value. This option is not valid without the -remove option.

Usage

The directory modify operation adds, removes, or changes a directory’s attributes and their values. The argument is a list of one or more names of directories to be operated on. Attribute options are not supported; use one or more of -add, -remove or -change, each of which use an attribute list as an argument.

Usually, you use the -remove option to remove a value from an attribute. You can use the -types option along with the -remove option to remove an entire attribute or list of attributes.

Some attributes in CDS are multi-valued. For instance, the CDS_Repos attribute can specify the locations and names several clearinghouses that maintain copies of a directory. The -add operation needs instructions whether it operates on single-valued or multi-valued attributes. Multi-valued attributes are the default case and are specified by using no qualifying options. However, you can specify the use of single-valued attributes by using the -single option.

Most attributes are usually managed by the client application. See the [z/OS DCE Administration Guide] for more information about attributes. All modifications are made to each directory listed in the argument.

Chapter 2. DCE Administration Commands 151
directory modify

error in any one causes the command to immediately end and generate an error. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission to the directory.

Examples

The following command sets the CDS_Convergence attribute on the /./depts/radiology directory to a value of low:

dcecp> directory modify /./depts/radiology -change {CDS_Converge low}
dcecp>

To add the value ontario to the attribute myname of a directory named /./sales, read the cds_attributes file (/opt/dcelocal/etc/cds_attributes) to verify that the attribute shown in the following display exists:

<table>
<thead>
<tr>
<th>OID</th>
<th>LABEL</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.22.1.3.91</td>
<td>myname</td>
<td>char</td>
</tr>
</tbody>
</table>

Enter the following command to assign the value ontario to the attribute myname:

dcecp> directory modify /./sales -add {myname ontario}
dcecp>

To remove the value 1 from the user-defined, set-valued attribute dirregion of a directory named /./sales, follow these steps:

1. Read the cds_attributes file (/opt/dcelocal/etc/cds_attributes) to check that the attribute dirregion is listed, as shown in the following display:

<table>
<thead>
<tr>
<th>OID</th>
<th>LABEL</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.22.1.3.66</td>
<td>dirregion</td>
<td>small</td>
</tr>
</tbody>
</table>

2. Enter the following command to remove the value 1 from the attribute dirregion:

dcecp> directory modify /./sales -remove {dirregion 1}
dcecp>

Related Information

Commands:
cdsccache clearinghouse link
cdscp dcecp object
directory operations

Returns a list of the operations supported by the directory object.

Format

directory operations

Usage

The directory operations command uses no arguments, and returns a list of the available operations for the directory object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the directory operations command.

Examples

dcecp> directory operations
add create delete list merge modify remove show synchronize help operations
dcecp>

Related Information

Commands:
cdscache clearinghouse link
cdscp dcecp object
**directory remove**

Deletes a child pointer from the directories specified.

**Format**

```plaintext
directory remove directory_name_list -member child_pointer_list
```

**Options**

- **-member child_pointer_list**
  
  This required option names one or more child pointers being removed from each directory in the operation argument.

**Usage**

The `directory remove` operation deletes a child pointer from the directories specified. This command is only needed to delete a child pointer that accidentally remains after the child directory is deleted. Usually, child pointers are removed internally by CDS when deleting directories with the `directory delete` command.

The `directory_name_list` argument is a list of names of one or more directories being operated on. The required `-member` option allows you to list the child pointers, specified as relative directory names, to be removed from each directory in the `directory_name_list` argument. An empty string is returned upon successful completion.

**Privilege Required:** You must have **d (delete)** permission to the child pointer or **a (admin)** permission to the parent directory.

**Examples**

The following command deletes the child pointer that accidentally remains after the `/./sales/east` directory is deleted:

```plaintext
dcecp> directory remove ././sales -member east
```

**Related Information**

Commands:

```plaintext
  cdscache  clearinghouse  link
  cdscp      dcecp        object
```
directory show

Returns a list of attributes for the specified directories and optionally, their specified contents.

Format

directory show directory_name_list
[-member child_pointer_list | -clearinghouse clearinghouse_name [-replica]] [-schema]

Options

- **clearinghouse** clearinghouse_name
  The **clearinghouse** option names the clearinghouse where the named directory exists.
  This option may not be specified with the **-member** option.

- **member** child_pointer_list
  This option specifies a list of relative directory names of one or more child pointers in
  the directory specified by the operation argument. The returned list describes the child
  pointer information for the specified member stored in the specified directories. This
  option may not be combined with the **-replica** option or the **-clearinghouse** option.

- **replica**
  This option specifies that the directory shown is a replica of an existing directory. If you
  use the **-replica** option, you must specify a clearinghouse using the **-clearinghouse**
  option. This option may not be combined with the **-member** option.

- **schema**
  This option returns whether an attribute is single or multi-valued. This is specific to a
  directory, meaning that the same attribute can be single-valued on one directory and
  multi-valued on another directory.

Usage

The **directory show** operation returns a list of attributes for the specified directories and optionally, their
specified contents. The **directory_name_list** argument is a list of one or more names of directories being
operated on. When used without any options, this command returns the attributes associated with the
named directories. If more than one directory is specified, then all the arguments are grouped together in
one list. The order of the returned arguments is the alphabetic order of the object identifiers (OIDs) of
each attribute for each directory.

You can request attributes of specific replicas in specific clearinghouses using the **-replica** and
**-clearinghouse** options. Or, you can request attributes of child pointers by using the **-member** option.

**Privilege Required:** You must have **r** (read) permission to the directories named in the argument
list.

Examples

The following example shows the attributes for the **/./depts/radiology** directory:
The following example uses the `schema` option to show whether the attributes for the `./:depts/radiology` directory are single-valued or multi-valued:

dcecp> directory show ./:depts/radiology -schema
{RPC_ClassVersion multi}
{CDS_CTS single}
{CDS_UTS single}
{CDS_ObjectUUID single}
{CDS_Replicas multi}
{CDS_AllUpTo single}
{CDS_Convergence single}
{CDS_ParentPointer multi}
{CDS_DirectoryVersion single}
{CDS_ReplicaState single}
{CDS_ReplicaType single}
{CDS_LastSkulk single}
{CDS_LastUpdate single}
{CDS_Epoch single}
{CDS_ReplicaVersion single}
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>clearinghouse</th>
<th>link</th>
<th>cdscp</th>
<th>dcecp</th>
<th>object</th>
</tr>
</thead>
</table>
directory synchronize

Initiates an immediate skulk of the directories specified.

Format

directory synchronize directory_name_list

Usage

The directory synchronize operation starts an immediate skulk of the directories specified. The directory_name_list argument is a list of names of one or more directories to be operated on. Skulks begin immediately in sequence. The command does not return until all skulks complete. An empty string is returned upon successful completion.

Privilege Required: You must have a (admin), w (write), i (insert), and d (delete) permissions to the directory. The server principal needs a (admin), r (read), and w (write) permissions to the directory.

Examples

The following command begins a skulk on the /.:admin directory:

dcecp> directory synchronize /.:admin

dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>clearinghouse</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>dcecp</td>
<td>object</td>
</tr>
</tbody>
</table>
**dts**

A **dcecp** object that manages a DTSD (Distributed Time Services daemon) process.

**Format**

- `dts activate [dts_server] [-abruptly]`
- `dts catalog [cellname] [-simplename]`
- `dts configure [dts_server] [-global | -notglobal]`
- `dts deactivate [dts_server]`
- `dts help [operation | -verbose] [-syntax]`
- `dts modify [dts_server] {<change attribute_list | attribute options}]
- `dts operations`
- `dts show [dts_server] [-attributes] [-counters] [-all]`
- `dts stop [dts_server]`
- `dts synchronize [dts_server] [-abruptly]`

**Parameters**

- `cellname`  The **dts catalog** operation uses an optional `cellname` argument that can return DTS servers registered in a foreign cell. The format is:
  
  /.../foreign_cellname
  
  An example is:
  
  /.../their_cell.goodco.com

- `dts_server`  Some **dts** operations take one optional argument that can be the name or the binding of one **dtsd** server to act on. The name syntax is:
  
  /.../cellname/hosts/hostname/dts-entity
  
  The **dtsd** can also be specified with a string binding for the remote host on which the **dtsd** is running. Use a string binding such as
  
  ncacn_ip_tcp:130.105.1.227
  
  Or, you can specify the binding using a **dcecp** string syntax such as:
  
  {ncacn_ip_tcp 130.105.1.227}

- `operation`  The name of one specific **dts** operation (subcommand) about which you want help information.

**Usage**

The **dts** object represents the **dtsd** process running on a host. The DTS process does not maintain stored data as some other objects do. Therefore, the **dts** object represents the information in and about a process rather than some data stored somewhere as other objects do.

These commands all affect the local **dtsd** entity by default. If given an argument, they can affect a remote DCE **dtsd**. The argument is a single server entry or string binding representing a **dtsd** that is contacted for the operation. If the _s(dts) convenience variable is set, it is treated as the name of a **dtsd** to contact for subsequent operations. If either of these methods is used, the specified server is the only server that
is contacted in an attempt to complete the operation. The argument on the command line takes precedence over the value of the _s(dts) convenience variable. These commands do not set the value of this variable after completion.

There are a number of attributes associated with the dts object. All of these can be viewed with the show operation, and many can be changed with the modify operation. Attribute arguments can contain a maximum of 80 characters and are recalculated to a normalized date format. For example, if the input value is 0-0025:10:99.99999999, the result is 1-01:11:39.990.

Note: Before the dcecp dts commands can be used, the dtscp create command must be used to create a DCE DTS entity on the system.

Attributes

The dts object supports attributes and counters. Most attributes and counters pertain to dtgd processes in general. A subset of attributes and counters pertains only to dtgd processes that are enabled as DTS server entities.

**autotdfchange {yes | no}**

Specifies whether automatic changes to the time differential factor are enabled or disabled. The value is either yes or no. The value is determined by the operating system (that is, it cannot be changed with the modify operation).

**clockadjrate**

Specifies the rate at which the DTS server or clerk entity adjusts the node clock during a synchronization. This may not be set by a user, but is built into dtgd.

**clockresolution**

Specifies the amount of time between system clock ticks. The value is determined by the operating system (that is, it cannot be changed with the modify operation).

**globalservers relative-time**

Specifies the set of global servers known by the node. The value cannot be changed with the modify operation. The information returned for each server is as follows:

- The DCE name of the host followed by /self
- The last time polled, the last observed time
- The last observed skew
- A binary value of whether the server was used in the last synchronization
- The transport time

The sub-attributes are respectively called:

- name
- timelastpolled
- lastobstime
- lastobsskew
- inlastsync
- transport

**globaltimeout relative-time**

Specifies the amount of time the node waits for a response to a WAN synchronization request before sending another request or declaring a global server as unavailable. The number of attempts made to reach the server is controlled by the queryattempts attribute. The default value is 0-00:01:30.000, and the range of possible values is 0-00:00:00.000 - 0-00:10:00.000.

**localservers**

Specifies the set of local servers known by the node. The value cannot be changed with the modify operation. The information returned for each server is as follows:

- The principal name that the server is running as
The last time polled
The last observed time
The last observed skew
A binary value of whether the server was used in the last synchronization
The transport time

The sub-attributes are respectively called:

- **name**
- **timelastpolled**
- **lastobstime**
- **lastobsskew**
- **inlastsync**
- **transport**

**localtimeout** `relative-time`

Specifies the amount of time the node waits for a response to a synchronization request before sending another request or declaring a server as unavailable. The number of attempts made to reach the server is controlled by the `queryattempts` attribute. The default is `00:01:00.000`, and the range of possible values is `00:00:00.000 - 00:01:00.000`.

**Note:**
This attribute controls only the initial contact with a time provider. During this initial contact, the time provider itself determines the timeout value for actually reporting times. This allows a time provider attached to a slow source like a modem to request that `dtsd` wait for a longer interval.

**maxdrift**

Specifies the worst-case drift rate of the node clock, in nanoseconds per second, as determined by the manufacturer specifications (that is, it cannot be changed with the `modify` operation).

**maxinaccuracy** `relative-time`

Specifies the inaccuracy limit for the node. When the node exceeds the maximum inaccuracy setting, it attempts to synchronize. The default is `00:00:00.100`, and the range of possible values is `00:00:00.000 - 1675199:02:48:05.478`.

**Note:**

The maximum number of hours is 24. A practical value is less than 60 seconds.

**minservers** `integer`

Specifies the minimum number of servers required for a synchronization. Settings of 1 or 2 may cause unreliable computed times. The default is 3, and the range of possible values is 1-10.

**nexttddfchange**

Specifies the future time at which the time differential factor is automatically changed. The value is determined by the operating system (that is, it cannot be changed with the `modify` operation).

**queryattempts** `integer`

Specifies the number of attempts that a node makes to contact a server before the node considers the server unavailable. The default is 3, and the range of possible values is 1-10.

**syncinterval** `relative-time`

Specifies the interval a node must wait to synchronize. Also specifies synchronization frequency when a node reaches the value specified by the `maxinaccuracy` attribute. For clerks, the default is `00:10:00.0`, and the range of possible values is `00:00:30.0`
- 01 00:00:00.00. For servers, the default 00:02.00.0, and the range of possible values is 00:00:30 - 01 00:00:00.00.

tdf relative-time Specifies the Time Differential Factor (TDF), which is the amount of time the server varies from Greenwich mean time (GMT) or Coordinated Universal Time (UTC). The default is 00:00:00.000, and the range of possible values is -13-00:00:00 - 13-00:00:00. This is obtained from various time zone information repositories; for example, the TZ environment variable, kernel structures, and so on, but may not be set by a user.

timerep Specifies the internal timestamp format used by the node. This is not related to the format that displays the current time to the user (see the clock show command below). DTS uses only V1.0 timestamps. This attribute may not be set by the user; its value is built into a dtsd.

tolerance relative-time Specifies the maximum separation allowed between the local clock and the computed time before synchronizations become abrupt rather than gradual (monotonic). The default is 00:05:00.000, and the range of possible values is 00:00:00.500 - 10675199-02:48:05.478.

type Specifies whether the node is a DTS server or clerk. This attribute cannot be set by the user.

version Specifies the DTS software version installed on the node. This attribute cannot be changed with the modify operation.

DTS Server Attributes

actcourierrole Specifies a server's acting interaction with the set of global servers. The values are the same as for the courierrole attribute below. When courierrole is set to backup, actcourierrole identifies whether this server is currently acting as the courier. This attribute cannot be set by the user.

checkinterval Specifies the amount of time between checks for faulty servers. This is applicable only to servers that have external time providers. The default is 01:30:00.00, and the range of the possible values is 00:00:30.000 - 10675199-02:48:05.478.

courierrole Specifies a server's interaction with the set of global servers. Possible values are:

backup The local server becomes a courier if none are available on the LAN. This is the default.

courier The local server synchronizes with the global set of servers.

noncourier The local server does not synchronize with the global set of servers.

ePOCH Specifies the server's epoch number. The default is 0, and the range of possible values is 0-255. This value may not be changed using the modify command; use the clock set command with the -abruptly and -epoch options to change its value.

lastsync Specifies the computed time at the start of the last attempted synchronization.

provider Specifies whether the entity used an external time-provider at the last successful synchronization. This attribute applies to servers only and may not be set by a user. The value is either yes or no.

serverentry Specifies a server's ACL entry name. The default setting is the following recommended value:

hosts/hostname/dts-entity
servergroup Specifies the security group name for the time servers within the cell. The default is:

```
servergroup
```

subsys/dce/dts-servers

description Specifies a server principal name for authentication purposes. serverprincipal is a read-only attribute. The default setting is the following recommended value:

```
description
```

hosts/hostname/self

status Specifies the state of the DTS entity. This is a read-only attribute and its possible values are:

- **disabled** The DTS entity is disabled.
- **enabled** The DTS entity is enabled.
- **syncing** The DTS entity is synchronizing.
- **updating** The DTS entity is updating the time.

uuid uid Specifies the unique identifier of the entity, which is generated when the entity is created.

General Counters

abrupts Specifies the number of times the node clock was set non-monotonically (abruptly).

badlocalservers Specifies the number of times that a local server was contacted, but it was not in the DTS security group.

badprotocols Specifies the number of times the local node failed to process a received message containing an incompatible protocol version.

badtimeres Specifies the number of times the local node failed to process a received message containing an incompatible timestamp format.

creationtime Specifies the time at which the DTS entity was created and the counters were initialized.

disables Specifies the number of times the DTS was disabled.

enables Specifies the number of times the DTS was enabled.

nointersections Specifies the number of times the node time interval failed to intersect with the computed interval of the servers.

nomemories Specifies the number of times the node was unable to allocate virtual memory.

providertimeouts Specifies the number of times a dtsd server process began contact with a time provider and did not receive the initial response within the interval specified by the localtimeout attribute.

syncs Specifies the number of times the node successfully synchronized.

syserrors Specifies the number of times a DTS process detected a system error.

toofewservers Specifies the number of times a node failed to synchronize because it did not contact the required minimum number of servers.

DTS Server Counters

badservers Specifies the number of times that a non-local server was contacted, but it was not in the DTS security group.

diffepochs Specifies the number of times the node received time response messages from servers or clerks with epoch numbers different from its own.
epochchanges  Specifies the number of times the server epoch changed.
faultyservers  Specifies the number of times a server detected faulty servers (other than itself).
noglobals  Specifies the number of times the courier server did not contact any global servers.
noresponses  Specifies the number of times the courier server did not contact a specific global server.
providerfailures  Specifies the number of times the external time-provider signaled a failure or the node was unable to access the time-provider.
syncattempts  Specifies the number of times a server attempted to synchronize its clock.

See the [z/OS DCE Administration Guide](#) for more information about attributes.

**Related Information**

Commands:

```
clock       dcecp       dtscp       utc
```
dts activate

Changes a DTS entity from an inactive state to an active state.

Format

dts activate [dts_server] [-abruptly]

Options

-abruptly Specifies to set the clock abruptly rather than gradually adjust it to the computed time.

Usage

The dts activate operation changes a DTS entity from an inactive state to an active state. The status attribute is changed to enabled. This tells the DTS entity to begin synchronizing. dts activate uses an -abruptly option to determine if the first clock adjustment as a result of synchronization is an abrupt or gradual one. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity

Examples

The following example activates a dtsd on the local host:

dcecp> dts activate
dcecp>

The following example activates a dtsd on a remote host named cyclops:

dcecp> dts activate /.:/hosts/cyclops/dts-entity
dcecp>

Related Information

Commands:

clock       dcecp       dtscp       utc
dts catalog

Returns a list of the names of all DTS servers registered in the LAN profile.

Format

dts catalog [cellname] [-simplename]

Options

-simplename  Returns a list of registered dts server names that are relative, not fully qualified.

Usage

The dts catalog operation returns a list of the names of all DTS servers registered in the LAN profile, /.:/lan-profile. The operation uses an optional cellname argument that can return the names of DTS servers registered in a foreign cell. By default, fully qualified names are returned in the form /.../cellname/hosts/hostname/dts-entity. If the -simplename option is used, the cell name is relative, not fully qualified. Names are returned in alphabetic order.

Privilege Required:  You must have r (read) permission to the cell root (/.) directory and to the LAN profile (/:.:/lan-profile).

Examples

dcecp> dts catalog
/.../mycell.goodco.com/hosts/cyclops/dts-entity
/.../mycell.goodco.com/hosts/medusa/dts-entity
/.../mycell.goodco.com/hosts/ninja/dts-entity

dcecp>

Related Information

Commands:
clock           dcecp           dtscp           utc
dts configure

Configures the local dtsd as a local or global server.

Format

dts configure [dts_server] {-global | -notglobal}

Options

global  Configures the system as a global server by adding the server entry to the cell profile.
notglobal  Configures the system as a local server by removing the server entry from the cell profile.

Usage

The dts configure operation sets the local dtsd as a local or global server. Requires one of two options, -global or -notglobal, to configure the local dtsd as a global server or not. The difference is whether they are listed in /./cell-profile. Returns the string global or notglobal to specify the current (new) state of the dtsd.

Privilege Required:  You must have w (write) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity

Examples

The following example sets the local dtsd as a global DTS server:

dcecp> dts configure -global
global
dcecp>

Related Information

Commands:
clock
dtscp
dcecp
utc
**dts deactivate**

Changes a DTS entity from an active state to an inactive state.

**Format**

*dts deactivate [dts_server]*

**Usage**

The *dts deactivate* operation changes a DTS entity from an active state to an inactive state. The *status* attribute is changed to *disabled*. This tells the DTS entity to stop synchronizing. An empty string is returned upon successful completion.

**Privilege Required:** You must have *w (write)* permission on the ACL associated with the DTS entity */.../cellname/hosts/hostname/dts-entity*

**Examples**

```
dcecp> dts deactivate
dcecp>
```

**Related Information**

Commands:

```
clock        dcecp        dtscp        utc
```
dts help

Returns help information about the dts object and its operations.

Format

dts help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each dts operation.
-verbose Displays information about the dts object.

Usage

The dts help command is used without an argument or option to return brief information about each dts operation.

Use the optional operation argument for additional information about the options that are valid for the operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the dts object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the dts help command.

Examples

dcecp> dts help
activate Activates a DTS entity.
catalog Returns a list of DTS servers in the cell.
configure Configures current dtsd as 'global' or 'notglobal'.
deactivate Deactivates a DTS entity.
modify Modifies attributes of the DTS entity.
show Displays attributes or counter info from the current dtsd.
stop Stops the current dtsd process.
synchronize Synchronizes the local dtsd with DTS servers.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:
clock dcecp dtscp utc
dts modify

Changes attributes of dtsd processes.

Format

dts modify [dts_server] {-change attribute_list \ attribute_options}

Options

-attribute_option attribute_value
   As an alternative to using the -change option with an attribute list, you can create
   individual attribute options by adding a hyphen (-) before any of the modifiable attributes
   listed in "Attributes" on page 159. The -attribute option is intended for use in scripts
   when you can paste in lengthy attribute lists output by previous commands. The
   individual attribute options may be easier to use for interactive commands.

-change attribute_list
   Lets you specify attributes using an attribute list. The format is:

   -change {{attr attr_value} {attr attr_value} ... {attr attr_value}}

Usage

The dts modify operation changes attributes of dtsd processes and allows attributes to be changed with
the -change option. Attribute options are also supported for all modifiable attributes. An empty string is
returned upon successful completion.

Privilege Required: You must have w (write) permission on the ACL associated with the DTS entity
/.../cellname/hosts/hostname/dts-entity

Examples

The following example sets the minimum number of servers needed for DTS operation to 2 for a remote
dtsd. (Generally, a DCE cell has a minimum of 3 DTS servers.)

dcecp> dts modify ncacn_ip_tcp:13wzerodot.1wzerodot5.1.227 -minservers 2

dcecp>

Related Information

Commands:
clock dcecp dtscp utc
dts operations

Returns a list of the operations supported by the dts object.

Format

dts operations

Usage

The dts operations command uses no arguments, and returns a list of the available operations for the dts object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the dts operations command.

Examples

dcecp> dts operations
activate catalog configure deactivate modify show stop synchronize help operations
dcecp>

Related Information

Commands:

clock dcecp dtscp utc
dts show

Shows attribute information for the specified dtsd processes.

Format

dts show [dts_server] [-attributes] [-counters] [-all]

Options

-all          Returns the attributes and counters for the local dtsd process.
-attributes   Returns only the attributes for the local dtsd process.
-counters    Returns only the counters for the local dtsd process.

Usage

The dts show operation shows attribute information for the specified dtsd processes. When called with the -attributes, an attribute list is returned with the values of the attributes listed above. If called with the -counters option, then counter information is returned. If called with the -all or with both the -attributes and -counters options, then both attribute and counter information is returned. The default behavior (started by using no options) is the same as if using the -attributes option. Attributes and counters are listed in the order they are returned by the server.

Privilege Required:  You must have r (read) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity
Examples

dcecp> dts show
{checkinterval +0-01:30:00.000I-----}
{epoch 0}
{tolerance +0-00:10:00.000I-----}
{tdf -0-05:00:00.000I-----}
{maxinaccuracy +0-00:00:00.100I-----}
{minservers 2}
{queryattempts 3}
{localtimeout +0-00:00:05.000I-----}
{globaltimeout +0-00:00:15.000I-----}
{syncinterval +0-00:02:00.000I-----}
{type server}
{courierrole backup}
{actcourierrole courier}
{clockadjrate 10000000 nsec/sec}
{maxdriftrate 10000000 nsec/sec}
{clockresolution 10000000 nsec}
{version V1.0.1}
{timerep V1.0.0}
{provider no}
{autotdfchange no}
{nexttdfchange 1994-10-30-01:00:00.000-05:00010.000}
{serverprincipal hosts/medusa/self}
{serverentry hosts/medusa/dts-entity}
{servergroup subsys/dce/dts-servers}
{status enabled}
{uuid 000013ed-000b-0000-b8ef-03a4fcdf09a4}
dcecp>

Related Information

Commands:

clock     dcecp     dtscp     utc
dts stop

Stops the dtsd process.

Format

dts stop [dts_server]

Usage

The dts stop operation stops the dtsd process. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity

Examples

The following example stops the dtsd process on the remote host named cyclops:

dcecp> dts stop ./:/hosts/cyclops/dts-entity

dcecp>

Related Information

Commands:

clock          dcecp          dtscp          utc
dts synchronize

Causes the dtsd to synchronize with DTS servers.

Format

dts synchronize [dts_server] [-abruptly]

Options

-abruptly Specifies to set the clock abruptly rather than gradually adjust it to the computed time.

Usage

The dts synchronize operation causes the dtsd to synchronize with DTS servers. The machine clock is adjusted accordingly. By default, the clock is adjusted gradually. This command is also used the optional -abruptly option to set the clock abruptly. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission on the ACL associated with the DTS entity /.../cellname/hosts/hostname/dts-entity

Examples

The following example causes the local dtsd process to synchronize with other DTS servers:

dcecp> dts synchronize

dcecp>

The following example causes the dtsd process on remote host named cyclops to synchronize immediately with other DTS servers:

dcecp> dts synchronize /.:/hosts/cyclops/dts-entity -abruptly

dcecp>

Related Information

Commands:

clock dcecp dtscp utc
endpoint

A dcecp object that manages endpoint information in local RPC endpoint maps.

Format

endpoint create {[-interface interface_id -binding protocol_sequence_list] [-object object_uuid_list]} [-annotation annotation][-noreplace]
endpoint delete {[-interface interface_id -binding protocol_sequence_list] [-object object_uuid_list]}
endpoint help [operation | -verbose] [-syntax]
endpoint operations
endpoint show [host_address] [[-interface interface_id [-version versions]] [-object object_uuid_list] | -uuid]
endpoint wlb_activate {[-interface interface_id -binding protocol_sequence_list] [-object object_uuid_list] [-activate act_boolean]}
endpoint wlb_ae_show [-aename {aename}]
endpoint wlb_off [{-aename {aename} [-all]}]
endpoint wlb_on [{-aename {aename} [-all]}]
endpoint wlb_show [host_address] [[-interface interface_id [-version versions]] [-object object_uuid_list] | -uuid]

Parameters

host_address  A protocol sequence identifying the host whose endpoint map is being shown.
operation     The name of one specific endpoint operation (subcommand) about which you want help information.

Data Structures

aename        Specifies the character string that the DCE RPC runtime uses to map groups of servers to the workload management (WLM) subsystem, in a parallel sysplex environment. A group of servers that have the same set of interfaces and that you want to be workload balanced should all have the same aename. aename is used with the endpoint wlb_ae_show, endpoint wlb_off, and endpoint wlb_on commands. This string can be up to 18 bytes.

interface_id  The interface identifier of an RPC interface. The interface identifier is in the following form:
interface-uuid,major-version.minor-version

The version numbers are optional, but if you omit a version number, the value defaults to 0. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:
-interface ec1eeb6Rzerodot5943-11c9-a3Rzerodot9-Rzerodot8Rzerodot2b1Rzerodot2989,3.11
Leading zeros in version numbers are ignored.
Or, you can use a dcecp string syntax:
interface-uuid major-version.minor-version
For example:
-interface {458ffcbe-98c1-11cd-bd93-000008adf56 1.0}

**protocol_sequence**

An RPC string binding that describes a server's location. The value has the form of an RPC string binding, without an object UUID. The binding information contains an RPC protocol, a network address, and sometimes an endpoint within brackets (rpc-prot-seq:network-addl[endpoint]). For a well-known endpoint, include the endpoint in the string binding surrounded by brackets. You may need to use the backslash character (\) to escape the brackets as shown in the following example. Otherwise, dcecp interprets the brackets as enclosing another command.

-binding ncadg_ip_udp:63.0.2.17\[5347\]

For a dynamic endpoint, omit the endpoint from the string binding. For example:

-b ncacn_ip_tcp:16.2.15.2

Or, you can use a dcecp string syntax. For example:

-binding {ncacn_ip_tcp 130.105.1.227 1072}

**object_uuid**

The UUID of an object. The UUID is a hexadecimal string. For example:

-object 3c6b8f60-5945-11c9-a236-08002b102989

Or, you can use a dcecp string syntax. For example:

-object {3c6b8f60-5945-11c9-a236-08002b102989}

**host_address**

An RPC string binding that describes the location of a host. The binding information contains an RPC protocol and the network address of the host.

**annotation**

An informational text string that helps you identify the purpose of the endpoint. Use quotation marks around the annotation field of endpoints to include internal spaces in an annotation. For example:

-annotation "Bulletin Board Server, Version 1.3a"

Or, you can use a dcecp string syntax. For example:

-annotation {Bulletin Board Server, Version 1.3a}

**versions**

Specifies interface version numbers being returned with an endpoint show operation. Specify versions using one of the following values for the -version option:

<table>
<thead>
<tr>
<th>Versions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>The interface version is ignored.</td>
</tr>
<tr>
<td>exact</td>
<td>Both the major and minor versions must match the specified versions.</td>
</tr>
<tr>
<td>compatible</td>
<td>The major version must match the specified version, and the minor version must be greater than or equal to the specified version.</td>
</tr>
<tr>
<td>major_only</td>
<td>The major version must match the specified version; the minor version is ignored.</td>
</tr>
<tr>
<td>upto</td>
<td>The major version must be less than or equal to that specified. If the major versions are equal, the minor version must be less than or equal to that specified.</td>
</tr>
</tbody>
</table>

If the -version option is absent, the command shows compatible version numbers.
Usage

The **endpoint** object operates on RPC endpoint mappings on the local host. Endpoints contain an interface identifier and one or more string bindings. Optionally they contain object UUIDs and an annotation.

Endpoint mappings are stored in the endpoint map maintained by the DCE daemon (**dced**) for z/OS DCE hosts.

MVS/ESA™ OpenEdition® DCE Release 1 uses the RPC daemon (**rpcd**) to maintain the endpoint map. The **server** object has some operations (for example, **disable** and **enable**) that affect endpoints maintained by the **dced**. However, **server** object operations do not operate on endpoints maintained by hosts based on OSF DCE level 1.0, such as those of MVS/ESA OpenEdition DCE Release 1. The **endpoint** object affects all endpoint maps, regardless of whether maintained by **rpcd** or **dced**.

Because endpoints have no names, the argument to these commands is not the name of an endpoint. Earlier versions of **rpccp** and **rpcd** allowed remote access to endpoints, but this was a security problem. Only the **endpoint show** operation allows access to endpoint maps on remote systems. The **server** object allows some remote operations on **dced** endpoint maps which are free of the security problem.

Use the various **endpoint** subcommands to create, delete, and show RPC endpoint information in local host endpoint maps.

Related Information

Commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>rpcentry</td>
<td>rpcprofile</td>
</tr>
<tr>
<td>rpccp</td>
<td>rpcgroup</td>
<td>server</td>
</tr>
</tbody>
</table>
endpoint create

Creates new endpoints in the local endpoint map database.

Format

endpoint create {[-interface interface_id -binding protocol_sequence_list] [-object object_uuid_list]} [-annotation annotation] [-noreplace]

Options

-annotation annotation
  Defines an annotation string for the endpoint. The annotation string enables you to identify the purpose of the endpoint. The annotation can be any textual information; for example, an interface name associated with the interface identifier or a description of a service or resource associated with a group.
  
  Use quotation marks around the annotation field of endpoints to include internal spaces in an annotation.

-binding protocol_sequence_list
  This required option declares a list of one or more protocol sequences (RPC string bindings) for the endpoint create operation.
  
  See "Data Structures" on page 175 for the format of a protocol sequence.

-interface interface_id
  This required option declares the interface identifier of an RPC interface. The endpoint create command operates on only one interface_id; if you enter more than one, the command ignores all but the last interface identifier.
  
  See "Data Structures" on page 175 for the format of the interface identifier.

-noreplace
  Use the -noreplace option when you want a host to run multiple instances of a server. Usually, when you add an interface-binding combination (a mapping) that already exists in an endpoint map, dcecp replaces the existing mapping with the new one. This behavior limits the number of server instances to one. Bypass this limitation by using the -noreplace option. Using this option can cause obsolete endpoints to accumulate in the endpoint map. Remove obsolete endpoints using the endpoint delete command.

-object object_uuid_list
  Declares the UUID of an object. Each endpoint create operation accepts a list of up to 32 object UUIDs. The UUID is a hexadecimal string. See "Data Structures" on page 175 for the format of the object UUID.

Usage

The endpoint create operation creates new endpoints in the endpoint map database on the local host. This command uses no arguments; it requires the -interface and -binding options or the object option. It accepts the -annotation and -noreplace options. The value of the -binding and -object options can be a list, but the others must be a single value. If the mapping already exists, it is replaced unless the -noreplace option is included.

This command creates a cross product from the -interface, -binding, and -object options and adds each element in the cross product as a separate registration in the local endpoint map. If you supply no object UUIDs, the corresponding elements in the cross product contain a nil object UUID. For example, if you
endpoint create

have an interface, if1, three bindings, b1, b2, and b3, and four object UUID, o1, o2, o3, and o4, the resulting 12 elements in the cross product are:

{if1,b1,o1} {if1,b1,o2} {if1,b1,o3} {if1,b1,o4}
{if1,b2,o1} {if1,b2,o2} {if1,b2,o3} {if1,b2,o4}
{if1,b3,o1} {if1,b3,o2} {if1,b3,o3} {if1,b3,o4}

An annotation string is part of each of these 12 elements, but is not shown for clarity.

An empty string is returned upon successful completion.

Privilege Required: You must have i (insert) permission to the endpoint map

\texttt{./hosts/\textit{host\_name}/config/epmap}

Examples

The following command adds an endpoint to the endpoint map of a local host. This example uses a backslash (\) to escape the brackets. Otherwise, \texttt{dcecp} interprets the brackets as enclosing another command.

\begin{verbatim}
dcecp> endpoint create -interface 458ffdcbe-98c1-11cd-bd93-000008ad56 \\ > -binding ncacn_ip_tcp:13wzerodot.1wzerodot5.1.227[1wzerodot67]\n
dcecp>
\end{verbatim}

This example uses the \texttt{dcecp} string syntax to create an endpoint in the local host endpoint map.

\begin{verbatim}
dcecp> endpoint create -interface {458ffdcbe-98c1-11cd-bd93-000008ad56,1.wzerodot} \\ > -binding {ncacn_ip_tcp 130.105.1.227[1067]} \\ > -object {76030c42-98d5-11cd-88bc-000008ad56} \\ > -annotation {Bulletin Board Server, Version 1.3a}

dcecp>
\end{verbatim}

Related Information

Commands:

\begin{verbatim}
dcecp rpccp rpcentry rpcgroup rpcprofile server
\end{verbatim}
**endpoint delete**

Deletes the specified endpoints from the local endpoint map database.

**Format**

```
endpoint delete {
  [-interface interface_id]
  [-binding protocol_sequence_list]
  [-object object_uuid_list]
}
```

**Options**

- **-binding protocol_sequence_list**
  This option declares a list of one or more protocol sequence (RPC string bindings) for the `endpoint delete` operation. See "Data Structures" on page 175 for the format of a protocol sequence.

- **-interface interface_id**
  This option declares the interface identifier of an RPC interface. The `endpoint delete` command operates on only one `interface_id`. If you enter more than one, the command ignores all but the last interface identifier. See "Data Structures" on page 175 for the format of the interface identifier.

- **-object object_uuid_list**
  Declares the UUID of an object. Each `endpoint delete` operation accepts a list of up to 32 object UUIDs. The UUID is a hexadecimal string. See "Data Structures" on page 175 for the format of the object UUID.

**Usage**

The `endpoint delete` operation deletes the specified endpoints from the endpoint map database. This command uses no arguments. It requires the `-interface` and `-binding` options or the `-object` option. The values of all but the `-interface` option may be lists. If the mappings do not exist, an error is generated.

This command creates a cross product from the `-interface`, `-binding`, and `-object` options and removes each element in the cross product from the local endpoint map. See the `endpoint create` operation for more details.

An empty string is returned upon successful completion.

**Privilege Required:** You must have d (delete) permission to the endpoint map `/./hosts/host_name/config/epmap`

**Examples**

The following command removes an endpoint object from the endpoint map of the local host. This example uses a backslash (\) to escape the brackets. Otherwise `dcecp` interprets the brackets as enclosing another command:

```
dcecp> endpoint delete -interface 458ffcbf-98c1-11cd-bd93-0000c08ad56,1.0
> -binding ncacn_ip_tcp:130.105.1.227\[1072\]
dcecp>
```

This example uses the `dcecp` string syntax to delete an endpoint from the local host endpoint map.
dcecp> endpoint delete -interface \{458ffcb8-98c1-11cd-bd93-0000c08ad16,1.0} \
> -binding \{ncacn_ip_tcp 130.105.1.227 1072}\n
dcecp>

**Related Information**

Commands:

- `dcecp`
- `rpccp`
- `rpcentry`
- `rpcgroup`
- `rpcprofile`
- `server`
endpoint help

Returns help information about the **endpoint** object and its operations.

**Format**

`endpoint help [operation | -verbose] [-syntax]`

**Options**

- **-syntax** Displays the syntax diagram of each **endpoint** operation.
- **-verbose** Displays information about the **endpoint** object.

**Usage**

The **endpoint help** command is used without an argument or option to return brief information about each **endpoint** operation.

Use the optional `operation` argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the `-verbose` option for more detailed information about the **endpoint** object itself.

Use the `-syntax` option to display the syntax diagram of each operation or of only the operation specified using the `operation` argument.

**Privilege Required:** No special privileges are needed to use the **endpoint help** command.

**Examples**

```
dcecp> endpoint help
create Creates RPC Endpoints for the specified interface.
delete Deletes a set of RPC Endpoints.
show Returns the RPC Endpoints for a specified interface.
wlb_activate Changes activate state of an endpoint.
wlb_ae_show Displays endpoint if AENAME matches.
wlb_off Turns off workload balancing if AENAME matches.
wlb_on Turns on workload balancing if AENAME matches.
wlb_show Displays AENAME and activate state along with endpoint.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>
```

**Related Information**

Commands:

```
dcecp           rpcentry            rpcprofile
rpccp           rpcgroup            server
```
**endpoint operations**

Returns a list of the operations supported by the `endpoint` object.

**Format**

`endpoint operations`

**Usage**

The `endpoint operations` command uses no arguments, and returns a list of the available operations for the `endpoint` object. The order of the elements is alphabetic, except that `help` and `operations` are listed last.

**Privilege Required:** No special privileges are needed to use the `endpoint operations` command.

**Examples**

```
dcecp> endpoint operations
create delete show help operations

dcecp>
```

**Related Information**

Commands:

```
dcecp  rpcentry  rpcprofile
rppcp   rpcgroup  server
```
endpoint show

Returns a list of information about endpoints for the local host or a remote host.

Format

endpoint show [host_address]

[[-interface interface_id [-version versions]] [-object object_uuid_list] -uuid]

Options

host_address Specifies an optional protocol sequence that identifies a remote host whose endpoint map you want to display. If you omit this option, the command displays the endpoint map of the local host.

See "Data Structures" on page 175 for the format of a host address.

-interface interface_id This option specifies the interface identifier of an RPC interface for which you want the endpoint mapping information. The endpoint show command operates on only one interface_id; if you enter more than one, the command ignores all but the last interface identifier.

See "Data Structures" on page 175 for the format of the interface identifier.

-object object_uuid_list Declares the UUID of an object. Each endpoint show operation accepts a list of up to 32 object UUIDs. The UUID is a hexadecimal string.

See "Data Structures" on page 175 for the format of the object UUID.

-uuid Specifies that the UUID of the endpoint map is being returned. It cannot be used with -interface, -version, or -object.

-version versions Specifies interface version numbers being returned with an endpoint show operation. Specify versions using one of the following values for the -version option:

all
exact
compatible
major_only
upto

See "Data Structures" on page 175 for the exact behavior of the version values.

Privilege Required:  No special privileges are needed to use the endpoint show command.

Usage

The endpoint show operation returns a list of information about endpoints in the endpoint map of a local or remote host. With no options specified, it returns all the local endpoint mappings. The -interface, -version, and -object options can be used so that only those endpoint mappings matching the supplied values are returned. The -object option accepts a list as a value; the others do not. If an argument is supplied, it is the address of the host whose endpoint map is being shown. If no argument is supplied, the endpoint map of the local host is used. See "Data Structures" on page 175 for the format of a host address.
If the -uuid option is specified, then the UUID of the specified host endpoint map is returned, rather than any information about the endpoints themselves. Each endpoint map is given a UUID at creation. If you know the current UUID of an endpoint map, you can delete any other stale UUIDs that may be in the RPC entry. If you specify the -uuid option, you cannot specify any other options.

Examples

The following example uses the dcecp string syntax to specify an interface to show local endpoint map information.

dcecp> endpoint show -interface {458ffcbbe-98c1-11cd-bd93-0000c08ad6f6 1.0}
{{object 76030c42-981d-11cd-88be-0000c08ad6f6}
 {interface {458ffcbbe-98c1-11cd-bd93-0000c08ad6f6 1.0}}
 {binding {ncacn_ip_tcp 130.105.1.227 1072}}
 {annotation {Bulletin Board Server, Version 1.3a}}}
dcecp>

The following command shows the endpoint objects in the local endpoint map that contain the specified interface identifier. This interface supports two object UUIDs on two protocol sequences:

dcecp> endpoint show -interface 257df1c9-c6d3-11ca-8554-08002b1c8f1f,1.0
{{object a57104f4-dfd0-11ca-b428-08002b1c8a62}
 {interface {257df1c9-c6d3-11ca-8554-08002b1c8f1f 1.0}}
 {binding {ncacn_ip_tcp 130.105.1.227 1040}}
 {annotation {cdsd [910]}}}

{{object a57104f4-dfd0-11ca-b428-08002b1c8a62}
 {interface {257df1c9-c6d3-11ca-8554-08002b1c8f1f 1.0}}
 {binding {ncadg_ip_udp 130.105.1.227 1163}}
 {annotation {cdsd [910]}}}

{{object b32648c6-928d-11cd-b4b5-0000c08ad6f6}
 {interface {257df1c9-c6d3-11ca-8554-08002b1c8f1f 1.0}}
 {binding {ncacn_ip_tcp 130.105.1.227 1042}}
 {annotation cds_clerkserver}}

{{object b32648c6-928d-11cd-b4b5-0000c08ad6f6}
 {interface {257df1c9-c6d3-11ca-8554-08002b1c8f1f 1.0}}
 {binding {ncadg_ip_udp 130.105.1.227 1168}}
 {annotation cds_clerkserver}}
dcecp>

The following command shows the UUID of the endpoint map on the host with the specified network address:

dcecp> endpoint show ncadg_ip_udp:130.105.1.227 -uuid
7273c754-e51c-11cd-bc0e-0000c08de054
dcecp>

Related Information

Commands:

| dcecp | rpccp | rpcentry | rpcgroup | rpcprofile | server |
**endpoint wlb_activate**

Changes the activation information associated with the endpoint workload management to match the activated information that a user types.

**Format**

```
endpoint wlb_activate {[-interface interface_id -binding protocol_sequence_list] [-object object_uuid_list] [-activate act_boolean]}
```

**Options**

- **-binding protocol_sequence_list**
  This option declares a list of one or more protocol sequence (RPC string bindings) for the `endpoint wlb_activate` operation.

- **-interface interface_id**
  This option declares the interface identifier of an RPC interface. The `endpoint wlb_activate` command operates on only one `interface_id`. If you enter more than one, the command ignores all but the last interface identifier.

- **-object object_uuid_list**
  Declares the UUID of an object. Each `endpoint wlb_activate` operation accepts a list of up to 32 object UUIDs. The UUID is a hexadecimal string.

- **-activate act_boolean**
  To activate workload balancing, use an `act_boolean` of any one of the following:
  - `1`
  - `yes`
  - `TRUE`
  - `true`
  - `T`
  - `t`

  To deactivate workload balancing, use an `act_boolean` of any one of the following:
  - `0`
  - `no`
  - `FALSE`
  - `false`
  - `F`
  - `f`

**Usage**

Use the `wlb_activate` command to set or reset the activate bit associated with workload balancing. Use this command if only one endpoint needs to be activated or deactivated. `wlb_activate` is similar to `endpoint delete` in that both verbs utilize the same input parameters as the criteria for selecting an endpoint object to manipulate. Usually, it is more convenient to use `wlb_on` or `wlb_off` if you want to turn workload balancing on or off for a class of servers with the same AENAME. The results obtained are not predictable when servers with the same AENAME have a non-uniform setting for their activate bits. In general, servers with the same AENAME should accomplish their workload balancing goals by having the activate bits of all of them set. If workload balancing is not needed, then no activate bits should be set.
DCED servers and DCECP commands prior to OS/390® Release 6 are unaware of workload balancing and this command. In general, be careful when trying to use these features. Also ensure that:

- Each system in the parallel sysplex subsystem is running OS/390 DCE Release 6 or higher (on an OS/390 or z/OS system configured into your sysplex)
- Each system in the parallel sysplex subsystem is part of the same DCE cell
- The workload management subsystem is running in goal mode.

Workload management concepts are explained more completely in z/OS MVS Programming: Workload Management Services, SA22-7619.

**Privilege Required:** You must have **d (delete)** permission to the endpoint map `./:/hosts/host_name/config/epmap`

**Examples**

```bash
dcecp> endpoint wlb_activate \
   -i {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0} \
   -b {ncacn_ip_tcp 9.130.79.227 4982} \
   -a false
```

**Related Information**

Commands:

- `dcecp`
- `rpcentry`
- `rpcprofile`
- `rpccp`
- `rpcgroup`
- `server`
endpoint wlb_ae_show

Displays the endpoint information for a specific aename.

Format

endpoint wlb_ae_show {-aename {aename}}

Options

- -aename aename

Specifies the character string that the DCE RPC runtime uses to map groups of servers to the workload management (WLM) subsystem, in a parallel sysplex environment. A group of servers that have the same set of interfaces and that you want be workload balanced should all have the same aename. This string can be up to 18 bytes.

Usage

Use the wlb_ae_show command to obtain filtered information about workload balanced endpoints having a specific AENAME. Using wlb_show to obtain filtered output requires entering a relatively large amount of information, such as the interface ID or UUID. It is more convenient to display information about a particular class of workload balanced interfaces using the wlb_ae_show command since only the servers with the AENAME you specify are displayed. In general, AENAMEs are more easily remembered and are more easily entered than interface IDs.

DCED servers and DCECP commands prior to OS/390 Release 6 are unaware of workload balancing and this command. In general, be careful when trying to use these features. Also ensure that:

- Each system in the parallel sysplex subsystem is running OS/390 DCE Release 6 or higher (on an OS/390 or z/OS system configured into your sysplex)
- Each system in the parallel sysplex subsystem is part of the same DCE cell
- The workload management subsystem is running in goal mode.

Workload management concepts are explained more completely in z/OS MVS Programming: Workload Management Services, SA22-7619.

Privilege Required: You must have d (delete) permission to the endpoint map /.:hosts/host_name/config/epmap

Examples
dcecp> endpoint wlb_ae_show -aename joe

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
  {binding {ncacn_ip_tcp 9.130.79.227 4982}}
  {AENAME joe}
  {ACTIVATED 1}
  {RESERVED 0}
  {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
  {binding {ncacn_ip_tcp 9.130.79.145 4982}}
  {AENAME joe}
  {ACTIVATED 1}
  {RESERVED 0}
  {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
  {binding {ncadg_ip_udp 9.130.79.227 5063}}
  {AENAME joe}
  {ACTIVATED 1}
  {RESERVED 0}
  {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
  {binding {ncadg_ip_udp 9.130.79.145 5063}}
  {AENAME joe}
  {ACTIVATED 1}
  {RESERVED 0}
  {annotation {WLM Test Server 1.0}}}

Related Information

Commands:

  dcecp       rpcentry
  rpccp       rpcgroup
  rpcprofile  server
**endpoint wlb_off**

Turns off workload balancing for a specific class of servers having the same AENAME.

**Format**

`endpoint wlb_off`

**Options**

`-aename aename`

Specifies the character string that the DCE RPC runtime uses to map groups of servers to the workload management (WLM) subsystem in a parallel sysplex environment. A group of servers that have the same set of interfaces and that you want be workload balanced should all have the same `aename`. This string can be up to 18 bytes.

**Usage**

Use this command to quickly turn off workload balancing for a class of servers having the same AENAME. Using `wlb_activate` to perform the same results requires several instances of the command, each requiring entering the interface ID and binding. The `wlb_off` command communicates with the local DCED and makes its modifications to the local endpoint map database.

You must specify `-aename aename` with `endpoint wlb_off`.

DCED servers and DCECP commands prior to OS/390 Release 6 are unaware of workload balancing and this command. In general, be careful when trying to use these features. Also ensure that:

- Each system in the parallel sysplex subsystem is running OS/390 DCE Release 6 or higher (on an OS/390 or z/OS system configured into your sysplex)
- Each system in the parallel sysplex subsystem is part of the same DCE cell
- The workload management subsystem is running in goal mode.

Workload management concepts are explained more completely in [z/OS MVS Programming: Workload Management Services](SA22-7619).

**Privilege Required:** You must have `d (delete)` permission to the endpoint map `/:/hosts/host_name/config/epmap`

**Examples**
endpoint wlb_off

dcecp> endpoint wlb_off -aename joe

dcecp> endpoint wlb_ae_show -aename joe

{{interface {b81f4124-e8da-11d0-a07c-008005ac34e8 1.0}}
 {binding {ncacn_ip_tcp 9.130.79.227 4982}}
 {AENAME joe}
 {ACTIVATED 0}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-008005ac34e8 1.0}}
 {binding {ncacn_ip_tcp 9.130.79.145 4982}}
 {AENAME joe}
 {ACTIVATED 0}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-008005ac34e8 1.0}}
 {binding {ncadg_ip_udp 9.130.79.227 5063}}
 {AENAME joe}
 {ACTIVATED 0}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-008005ac34e8 1.0}}
 {binding {ncadg_ip_udp 9.130.79.145 5063}}
 {AENAME joe}
 {ACTIVATED 0}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

Related Information

Commands:

dcecp
rpccp
rpcentry
rpcgroup
rpcprofile
server
endpoint wlb_on

endpoint wlb_on

Turns on workload balancing for a specific class of servers having the same AENAME.

Format

endpoint wlb_on

Options

-aename aename

Specifies the character string that the DCE RPC runtime uses to map groups of servers to the workload management (WLM) subsystem in a parallel sysplex environment. A group of servers that have the same set of interfaces and that you want be workload balanced should all have the same aename. This string can be up to 18 bytes.

Usage

Use this command to quickly turn on workload balancing for a class of servers having the same AENAME. Using wlb_activate to perform the same results requires several instances of the command, each requiring entering the interface ID and binding. The wlb_on command communicates with the local DCED and makes its modifications to the local endpoint map database.

You must specify -aename aename with endpoint wlb_on.

DCED servers and DCECP commands prior to OS/390 Release 6 are unaware of workload balancing and this command. In general, be careful when trying to use these features. Also ensure that:

- Each system in the parallel sysplex subsystem is running OS/390 DCE Release 6 or higher (on an OS/390 or z/OS system configured into your sysplex)
- Each system in the parallel sysplex subsystem is part of the same DCE cell
- The workload management subsystem is running in goal mode.

Workload management concepts are explained more completely in z/OS MVS Programming: Workload Management Services, SA22-7619.

Privilege Required: You must have d (delete) permission to the endpoint map

Examples
dcecp> endpoint wlb_on -aename joe

dcecp> endpoint wlb_ae_show -aename joe

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
 {binding {ncacn_ip_tcp 9.130.79.227 4982}}
 {AENAME joe}
 {ACTIVATED 1}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
 {binding {ncacn_ip_tcp 9.130.79.145 4982}}
 {AENAME joe}
 {ACTIVATED 1}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
 {binding {ncadg_ip_udp 9.130.79.227 5063}}
 {AENAME joe}
 {ACTIVATED 1}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
 {binding {ncadg_ip_udp 9.130.79.145 5063}}
 {AENAME joe}
 {ACTIVATED 1}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

Related Information

Commands:

dcecp rpccp rpcentry rpcgroup rpcprofile server
endpoint wlb_show

Displays the AENAME information and activate state of endpoints in addition to all of the other information already displayed by endpoint show.

Format

endpoint wlb_show [host_address]
[-interface interface_id [-version versions]] [-object object_uuid_list] | -uuid

Options

host_address  Specifies an optional protocol sequence that identifies a remote host whose endpoint map you want to display. If you omit this option, the command displays the endpoint map of the local host.

Note: Unlike the endpoint show command, the remote system must be running OS/390 DCE Release 6 or higher on a system configured into your sysplex.

-interfacing  interface_id
  Specifies the interface identifier of an RPC interface for which you want the endpoint mapping information. The endpoint wlb_show command operates on only one interface_id; if you enter more than one, the command ignores all but the last interface identifier.

-object  object_uuid_list
  Declares the UUID of an object. Each endpoint wlb_show operation accepts a list of up to 32 object UUIDs. The UUID is a hexadecimal string. See "Data Structures" on page 175 for the format of the object UUID.

-uuid
  Specifies that the UUID of the endpoint map is being returned. It cannot be used with -interface, -version, or -object.

-version  versions
  Specifies interface version numbers being returned with an endpoint wlb_show operation. Specify versions using one of the following values for the -version option:

  all
  exact
  compatible
  major_only
  upto

Usage

This command is identical to the endpoint show command in that it displays all of the information associated with the regular show command. In addition, it displays the AENAME and the activated bits in the same bracket-delineated format as the regular show command. Like the show command, this command can take the host_address option. When using a host_address, ensure that the remote host is also running OS/390 DCE Release 6 or higher.

DCED servers and DCECP commands prior to OS/390 Release 6 are unaware of workload balancing and this command. In general, be careful when trying to use these features. Also ensure that:

- Each system in the parallel sysplex subsystem is running OS/390 DCE Release 6 or higher (on an OS/390 or z/OS system configured into your sysplex)
Each system in the parallel sysplex subsystem is part of the same DCE cell.
The workload management subsystem is running in goal mode.

Workload management concepts are explained more completely in [z/OS MVS Programming: Workload Management Services](https://www.ibm.com) SA22-7619.

**Privilege Required:** You must have **d (delete)** permission to the endpoint map

`/:.hosts/host_name/config/epmap`

**Examples**

dcecp> endpoint wlb_show

...  
...  
...  

```
{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
 {binding {ncadg_ip_udp 9.130.79.227 5063}}
 {AENAME joe}
 {ACTIVATED 1}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

{{interface {b81f4124-e8da-11d0-a07c-08005acd34e8 1.0}}
 {binding {ncadg_ip_udp 9.130.79.145 5063}}
 {AENAME joe}
 {ACTIVATED 1}
 {RESERVED 0}
 {annotation {WLM Test Server 1.0}}}

...
...
...
```

**Related Information**

Commands:

- `dcecp`
- `rpcentry`
- `rpcprofile`
- `rppp`
- `rpcgroup`
- `server`

Chapter 2. DCE Administration Commands 195
A **dcecp** object that manages a group in the DCE Security Service.

### Format

- **group add** `group_name_list -member member_name_list`
- **group catalog** `[server_name_list] [-simplename]`
- **group create** `group_name_list [-attribute attribute_list | attribute options]`
- **group delete** `group_name_list`
- **group help** `[operation | -verbose] [-syntax]`
- **group list** `group_name_list [-simplename]`
- **group modify** `group_name_list`
  
  `([-[change attribute_list | attribute options] |`
  `-add extended_registry_attribute_list |`
  `-remove extended_registry_attribute_list [-types])`
- **group operations**
- **group remove** `group_name_list -member member_name_list`
- **group rename** `group_name -to new_group_name`
- **group show** `group_name_list [-xattrs] [-all]`

### Parameters

- **group_name**  
  The name of one registry group being renamed.

- **group_name_list**  
  A list of one or more names of groups to act on. The names can all be fully qualified group names as in:
  
  `/.../cellname/group_name`

  Or, they can all be cell-relative group names as in:
  
  `group_name`

  Cell-relative names refer to a group in the cell named in the `_s(sec)` convenience variable or in the default cell of the local host if the `_s(sec)` convenience variable is not set.

  Do not mix fully-qualified names and cell-relative names in a list. And do not use names of registry database objects that contain registry information about groups (group names must not begin with `./sec/group`).

- **operation**  
  The name of one specific **group** operation (subcommand) about which you want help information.

- **server_name_list**  
  A list of one or more names of registry servers to act on. The names can be fully qualified as in:
  
  `/.../cellname/subsys/dce/sec/registry_name`

  Or, they can be cell-relative names as in:
  
  `subsys/dce/sec/registry_name`
Usage

The group object represents registry groups. Groups are collections of principal names. Unless otherwise noted, all of the operations of this object use one argument, which is the name of the group to act on. It must be a group name, not the name of the registry database object that contains the registry information about the group (that is, the name must not begin with /:/sec/group/). Non-fully qualified names refer to a group in the cell described in _s(sec) or in the cell of the local host if _s(sec) is not set.

After this command runs, the _b(sec) convenience variable is set to the name of the server that it was bound to for the command. The value of the variable _s(sec) before the command, is treated as a hint; the server specified is contacted if it can service the request. A case where it cannot service the request is if a read-only registry was bound to, and the next command is a create command. In this case, the master registry is bound to automatically and the _b(sec) variable updated appropriately. The value of the variable is the name of the registry bound to in one of the formats specified as valid for the argument to the registry object.

Attributes

alias value {yes | no}

Used with the create and modify operations, the value of this attribute is either yes or no. Each group can have only one name, but may have one or more alias names. All these names refer to the same group and, therefore, the same UUID and gid. While aliases refer to the same group, they are separate entries in the registry database. Therefore, the instance name given to a group command can refer to either the primary name or an alias name of a group. The value of this attribute determines this.

fullname string

Used with the create and modify operations to specify the full name of the group being added to the registry. The value is a string. If it contains spaces, it must be in quotation marks or braces. If not entered, the full name defaults to the null string (that is, blank).

gid integer

Used with the create operation to specify the Group Identifier for the group. If this attribute is not present, then a gid is assigned to the group automatically.

This attribute cannot be changed with the modify operation.

inprojlist value {yes | no}

Used with the create and modify operations to include the group in the principal project list. The value for this option is either yes or no. If it is no, then members of this group do not acquire the access rights of this group. The default is no.

uuid hexadecimal number

With the create operation, this attribute is used only to “adopt” an orphaned UUID. Usually, the UUID for a new group is generated by the registry. In cases where data exists, tagged with a UUID of a group that was deleted from the registry, this attribute can be used on the create command to specify the old UUID for a new group. The UUID specified must be an orphan (a UUID for which no name exists in the registry). An error occurs if you specify a name that is already defined in the registry.

Neither the alias or gid attributes may be used with this attribute. Both the fullname and inprojlist attributes may be used.

This attribute cannot be changed with the modify operation.

See the z/OS DCE Administration Guide for more information about attributes.
Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>organization</th>
<th>registry</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>principal</td>
<td>rgyedit</td>
</tr>
</tbody>
</table>
group add

Adds members to a security group.

Format

group add group_name_list -member member_name_list

Options

-member member_name_list
   A list of one or more names of principals (local or foreign) being added to each group in
   the argument.

Usage

The group add operation adds a member to a group. The argument is a list of names of groups to add
members to. The value of the required -member option is a list of names of principals (local or foreign)
being added to each group in the argument. A peer-to-peer connection must exist with the foreign cell in
order to add a principal from that cell. The principals must exist, or the command returns an error. An
empty string is returned upon successful completion.

Privilege Required:  You must have rM (read, Member_list) permission on the target group and rg
(read, groups) permission on the principal being added.

Examples

dcecp> principal create P_Pestana
dcecp> group add users -member P_Pestana
dcecp>

Related Information

Commands:

account          organization          registry
dcecp            principal            rgyedit
group catalog

Returns a list of the names of all groups in the registry.

Format

group catalog [server_name_list] [-simplename]

Options

-simplename Returns a list of group names in the registry that are relative, not fully qualified.

Usage

The group catalog operation returns a list of the names of all groups in the registry database. This command uses an argument to specify the server to use. By default, fully qualified names are returned in the form cellname/groupname. If the -simplename option is given, then the group name is relative, not fully qualified.

Privilege Required: You must have r (read) permission to the /.:/sec/group directory.

Examples

dcecp> group cat
/.../mycell.goodco.com/nogroup
/.../mycell.goodco.com/system
/.../mycell.goodco.com/daemon
/.../mycell.goodco.com/uucp
/.../mycell.goodco.com/bin
/.../mycell.goodco.com/kmem
/.../mycell.goodco.com/mail
/.../mycell.goodco.com/tty
/.../mycell.goodco.com/none
/.../mycell.goodco.com/tcb
/.../mycell.goodco.com/acct-admin
/.../mycell.goodco.com/subsys/dce/sec-admin
/.../mycell.goodco.com/subsys/dce/cds-admin
/.../mycell.goodco.com/subsys/dce/dts-admin
/.../mycell.goodco.com/subsys/dce/cds-server
/.../mycell.goodco.com/subsys/dce/dts-servers
/.../mycell.goodco.com/users

dcecp>

Related Information

Commands:

account          organization          registry
                  principal          rgyedit
group create

Creates a new group in the registry database.

Format

group create group_name_list [-attribute attribute_list | attribute options]

Options

-attribute attribute_list

Lets you specify attributes using an attribute list rather than using the -alias, -gid, -uuid, -inprojlist, and -fullname options. The format is:

-attribute {{alias value} {gid integer} {uuid hex number} {inprojlist value} {fullname string}}

The -attribute option is intended for use in scripts when pasting in lengthy attribute list output by previous commands. The individual attribute options may be easier to use for interactive commands.

Note that you can specify extended registry attributes (ERAs) with the -attribute option. You cannot specify ERAs as individual attribute options.

The alternative attribute options are:

alias value {yes | no}

Specifies whether the group name is an alias. A value of yes or specifying no value sets the name to an alias. When setting an alias, you must also use the -gid option. A value of no or the absence of the -alias option does not set the name to an alias.

-gid integer

Specifies the Group Identifier for the group. If this option is not present, a gid is assigned to the group automatically.

 fullname string

Specifies the full name of the group being added to the registry. The value is a string. If it contains spaces, it must be in quotation marks or braces. If not entered, the full name defaults to the null string (that is, blank).

-inprojlist value {yes | no}

This option includes the group in the principal project list. The value for this option is either yes or no. If it is no, then members of this group do not acquire the access rights of this group. The default is no.

-uuid hexadecimal number

This option is used only to adopt an orphaned UUID. Usually, the UUID for a new group is generated by the registry. In cases where data exists, tagged with a UUID of a group that was deleted from the registry, this option can be used on the create command to specify the old UUID for a new group. The UUID specified must be an orphan (a UUID for which no name exists in the registry). An error occurs if you specify a name that is already defined in the registry.

Neither the -alias or -gid options may be used with this option. Both the -fullname and -inprojlist options may be used.
**Usage**

The `group create` operation creates a new group in the registry database. The argument is a list of names of groups being created. An empty string is returned upon successful completion. Options specify the attributes of the newly created group. All options are applied to all groups in the argument.

A maximum of 256 values are allowed for the `attribute_list`.

**Privilege Required:** You must have `i` (insert) permission to the `./sec/group` directory.

**Examples**

```
dcecp> group create users4 -attribute {fullname "temporary users"}
dcecp>
```

**Related Information**

Commands:

- account
- dcecp
- organization
- principal
- registry
- rgyedit
group delete

Deletes groups from the registry.

Format

\texttt{group delete group\_name\_list}

Usage

The \texttt{group delete} operation deletes groups from the registry. The argument is a list of names of groups being deleted. If a named group does not exist, an error is generated. An empty string is returned upon successful completion.

When a group is deleted, any accounts associated with the group are deleted, also. To preserve any accounts, first add the principals to a different group using the \texttt{group add} command. Then modify the principals' accounts to point to the new group, using the \texttt{account modify} command. You can then remove the members from the group using the \texttt{group remove} command.

Privilege Required: You must have \texttt{d} (delete) permission to the directory. \texttt{/:.sec/group}. You must have \texttt{rD} (read, Delete_object) permission on the group being deleted.

Examples

dcecp> group delete users4
dcecp>

Related Information

Commands:

\begin{tabular}{lll}
\texttt{account} & \texttt{organization} & \texttt{registry} \\
\texttt{dcecp} & \texttt{principal} & \texttt{rgyedit} \\
\end{tabular}
Returns help information about the **group** object and its operations.

**Format**

```
group help [operation | -verbose] [-syntax]
```

**Options**

- **-syntax**
  Displays the syntax diagram of each **group** operation.

- **-verbose**
  Displays information about the **group** object.

**Usage**

The **group help** command is used without an argument or option to return brief information about each **group** operation.

Use the optional **operation** argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the **-verbose** option for more detailed information about the **group** object itself.

Use the **-syntax** option to display the syntax diagram of each operation or of only the operation specified using the **operation** argument.

**Privilege Required:** No special privileges are needed to use the **group help** command.

**Examples**

```
dcecp> group help
add                     Adds a member to the named group.
catalog                Returns a list of all the names of groups in the registry.
create                 Creates a group.
delete                 Deletes a group.
list                   Returns all of the members of a group.
modify                 Changes the information about a group.
remove                 Removes a specified member from the named group.
rename                 Renames the specified group.
show                   Returns the attributes of a group.
help                   Prints a summary of command-line options.
operations             Returns the valid operations for command.
dcecp>
```

**Related Information**

**Commands:**

- **account**
- **organization**
- **registry**
- **dcecp**
- **principal**
- **rgyedit**
group list

Returns a list of the names of all members of a group.

Format

group list group_name_list [-simplename]

Options

-simplename Returns the list of group names in the registry that are relative, not fully qualified. The fully-qualified name is always returned for members in a foreign cell.

Usage

The group list operation returns a list of the names of all members of a group. The argument is a list of names of groups being operated on. If more than one group is listed, the names are concatenated on output. By default, fully qualified names are returned in the form $cellname/groupname$. If the -simplename option is given, the cell name is relative, not fully qualified. An empty string is returned upon successful completion.

Privilege Required: You must have r (read) permission to the group.

Examples

dcecp> group list none
/.../mycell.goodco.com/dce-ptgt
/.../mycell.goodco.com/dce-rgy
/.../mycell.goodco.com/krbtgt/mymcell.goodco.com
/.../mycell.goodco.com/cell_admin
/.../mycell.goodco.com/hosts/pmin17/self

dcecp>

Related Information

Commands:

account      organization      registry
-dcecp       principal          rgyedit
group modify

Changes attributes of groups.

Format

group modify  group_name_list
{[-change  attribute_list | attribute options] |  
-add  extended_registry_attribute_list |  
-remove extended_registry_attribute_list[-types]}

Options

-**add**
  Lets you add extended registry attributes that may be defined for your environment. You can specify the attributes being added as a list of one or more extended registry attributes. See the [[z/OS DCE Administration Guide]] for more information about extended registry attributes.

-**change**  attribute_list
  Lets you modify specific attributes using an attribute list rather than using the -**alias**, -**inprojlist**, and -**fullname** options. The format is

  -change  {{alias  value} {inprojlist  value} {fullname  string}}

  The -change option is intended for use in scripts when pasting in lengthy attribute lists output by previous commands. The individual attribute options may be easier to use for interactive commands. Note that you can modify ERAs by including them in an attribute_list. You cannot specify ERAs as individual attribute options.

-**remove**  extended_registry_attribute_list
  Lets you remove extended registry attributes that may be defined for your environment. You can specify the attributes being removed as a list of one or more extended registry attributes. See the [[z/OS DCE Administration Guide]] for more information about extended registry attributes.

-**types**
  Used with the -remove option to specify that the value of the -remove option is a list of attribute types. This means that the entire attribute is removed, not just a value. This option is not valid without the -remove option.

Alternative attribute options are:

-**alias**  value {yes | no}
  Specifies whether the group name is an alias. A value of yes (or specifying no value) sets the name to an alias. A value of no or the absence of the -**alias** option does not set the name to an alias.

-**fullname**  string
  Specifies the full name of the group being added to the registry. The value is a string. If it contains spaces, it must be in quotation marks or braces. If not entered, the full name defaults to the null string (that is, blank).

-**inprojlist**  value {yes | no}
  Includes the group in the principal's project list.
Usage

The `group modify` operation changes attributes of groups. The argument is a list of names of groups being operated on. All modifications are applied to all groups named in the argument. Groups are modified in the order they are listed and all modifications to an individual group are atomic. Modifications to multiple groups are not atomic. A failure for any one group in a list generates an error, and the rest of the operation is stopped. An empty string is returned upon successful completion.

The `-change` option can be used to modify the value of any one of the attributes except for `gid` and `uuid`. The value of the `-change` option is an attribute list describing the new values for the specified attributes. `group modify` also supports the `-alias`, `-inprojlist`, and `-fullname` options. Specify only ERA attributes when using the `-add` and `-remove` options.

A maximum of 256 values are allowed in the `extended_registry_attribute_list`.

Privilege Required: You must have `rm (read, mgmt_info)` permission to the group being modified.

Examples

dcecp> `group modify users3 -change {fullname "General Nursing Staff"}`
dcecp> `group show users3`
{alias no}
{gid 5212}
{uuid 0000145c-9363-21cd-a601-0000c08adf56}
{inprojlist no}
{fullname {General Nursing Staff}}
dcecp>

Related Information

Commands:

account organization registry
dcecp principal rgyedit
group operations

Returns a list of the operations supported by the group object.

Format

group operations

Usage

The group operations command uses no arguments, and returns a list of the available operations for the group object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the group operations command.

Examples

dcecp> group operations
add catalog create delete list modify remove rename show help operations
dcecp>

Related Information

Commands:

account  organization  registry
dcecp       principal       rgyedit
group remove

Removes a member from a group.

Format

```
group remove group_name_list -member member_name_list
```

Options

- `member member_name_list`
  - A list of one or more names of principals (local or foreign) to add to each group in the argument.

Usage

The `group remove` operation removes a member from a group. The argument is a list of names of groups to remove members from. The value of the required `-member` option is a list of names of principals (local or foreign) to be removed from the groups listed in the argument. A peer-to-peer connection must exist with the foreign cell in order to add a principal from that cell. An empty string is returned upon successful completion.

When a member is removed from a group, any accounts associated with that principal and group are deleted. Accounts are associated with a principal, a group, and an organization, therefore, any accounts where both the principal name and group name match those given to this command are removed, but accounts where only one matches are untouched. To preserve any accounts, first add the principals to a different group using the `group add` command. Then modify the principals' accounts to point to the new group, using the `account modify` command. You can then remove the members from the group using the `group remove` command.

**Privilege Required:** You must have rM (read, Member_list) permission on the target groups and r (read) permission on the member being removed.

Examples

```
dcecp> group remove -member P_Pestana users
dcecp>
```

Related Information

Commands:

```
account          organization          registry
dcecp            principal            rgyedit
```
group rename

Changes the name of a specified group.

**Format**

```
group rename group_name -to new_group_name
```

**Options**

- `-to new_group_name`
  Specifies the new name of the group.

**Usage**

The `group rename` operation changes the name of a specified group. The argument is a single name of a group being renamed. This command uses a required `-to` option with a value of the new name. The value cannot be a list. An empty string is returned upon successful completion.

**Privilege Required:** You must have `rf (read, fullname)` permission to the specified groups.

**Examples**

```
dcecp> group rename users4 -to users_temporary
```

**Related Information**

Commands:

- `account`
- `organization`
- `registry`
- `dcecp`
- `principal`
- `rgyedit`
group show

Shows registry information for the specified groups.

Format

group show group_name_list [-xattrs] [-all]

Options

-**all** Returns fixed attributes and any extended registry attributes.
-**-xattrs** Returns only the extended registry attributes.

Usage

The *group show* operation returns an attribute list describing the specified groups. The argument is a list of names of groups being operated on. If more than one group is given, the attributes are concatenated. Attributes are returned in the following order:

- alias
- gid
- uuid
- inprojlist
- fullname

If called with the **-xattrs** option, the command returns Extended Registry Attributes instead of the above attributes. If called with **-all**, then both are returned.

Privilege Required: You must have r (read) permission to the specified groups.

Examples

dcecp> group show users_temporary
{alias no}
{gid 5211}
{uuid 0000145b-9362-21cd-a601-0000c08adf56}
{inprojlist no}
{fullname {temporary users}}
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>organization</th>
<th>registry</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>principal</td>
<td>rgyedit</td>
</tr>
</tbody>
</table>
host

A **dcecp** task object that allows administrators to easily start and stop host servers and view host information in a DCE cell.

**Format**

```
host catalog [cell_name] [-simplename]
host help [operation | -verbose] [-syntax]
host operations
host ping host_name
host show [host_name]
host start [host_name]
host stop [host_name] [-force]
```

**Parameters**

- `cell_name`: The name of a single cell to operate on. The name must be a fully qualified cell name (as in `/:hp2.../their_cell.goodcompany.com/`).
- `host_name`: A name of a single host to operate on. Some host commands accept both fully qualified names (as in `/..../their_cell.goodcompany.com/hosts/hostname/`) and cell relative names (as in `hosts/hostname/`), while others accept only fully qualified names. See the individual host command descriptions for details.
- `operation`: The name of one specific host operation (subcommand) about which you want help information.

**Usage**

The host object represents DCE processes running on a machine in (or being added in) a DCE cell. The host object allows administrators to easily start and stop DCE servers on machines.

The host object cannot start or stop the core DCE services on a machine running z/OS DCE. This includes `dced`, `cdsadv`, `dtsd`, and `auditd`. It can start or stop these servers on other machines. The argument to this command is the DCE name of a host to operate on. If an argument is omitted, the command operates on the local host, when possible. The behavior of commands operating locally may differ slightly with more operations performed on the local host than is possible remotely. See the individual host commands for details.

This object is implemented as a script to allow modifying and extending on a per-site basis. For example, administrators may want to add GDS and DFS information to the object.

**Related Information**

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>dcecp</th>
<th>directory</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>aud</td>
<td>dts</td>
<td>registry</td>
<td></td>
</tr>
</tbody>
</table>
**host catalog**

Returns a list of names of hosts in the cell.

**Format**

`host catalog [cell_name] [-simplename]`

**Options**

- `simplename` Specifies that the names of hosts in the cell are cell relative, not fully qualified.

**Usage**

The `host catalog` operation returns a list of names of hosts in the cell. The names are fully qualified unless the `simplename` option is used, in which case they are cell relative. The optional argument can specify a cell to operate in.

**Privilege Required:** You must have `r` (read) permission to the `/.:/hosts` directory in CDS.

**Examples**

The following example lists the full names all of the DCE hosts that have entries in the CDS `/./hosts` directory:

```
dcecp> host catalog
/.../mycell.goodco.com/hosts/alpha
/.../mycell.goodco.com/hosts/beta
/.../mycell.goodco.com/hosts/gamma
```

The following example lists all of the DCE hosts that have entries in the CDS `/./hosts` directory:

```
dcecp> host catalog -simplename
hosts/alpha
hosts/beta
hosts/gamma
```

**Related Information**

Commands:

- `account`
- `dcecp`
- `dts`
- `server`
- `directory`
- `registry`
host help

host help

Returns help information about the host object and its operations.

Format

host help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each host operation.
-verbose Displays information about the host object.

Usage

The host help command is used without an argument or option to return brief information about each host operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the host object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the host help command.

Examples

dcecp> host help
catalog Returns a list of configured hosts in the cell.
ping Determines if DCE is responding on the specified host.
show Returns a list of all DCE processes configured on the specified host.
start Starts DCE on the specified host.
stop Stops DCE on the specified host.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

account dcecp dts server
aud directory registry
host operations

Returns a list of the operations supported by the host object.

Format

host operations

Usage

The host operations command uses no arguments, and returns a list of the available operations for the host object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the host operations command.

Examples

dcecp> host operations
catalog ping show start stop help operations
dcecp>

Related Information

Commands:

account

aud
dcecp
directory
dts
registry

server
host ping

Tests whether DCE processes listed in the server configuration data are accessible from the network.

Format

host ping host_name

Usage

The host ping operation tests if DCE processes that are listed in the server configuration data are accessible from the network. It contacts the endpoint mapper on the specified host. The argument is the fully-qualified name of a single host to ping. Returns a 1 if the host responds, a 0 if not.

Privilege Required: No special privileges are required for the host ping command.

Examples

The following example pings host hydra:

dcecp> host ping /.:/hosts/hydra
1
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>dcecp</th>
<th>dts</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>aud</td>
<td>directory</td>
<td>registry</td>
<td></td>
</tr>
</tbody>
</table>
host show

Returns a list describing the processes listed in the server configuration data for the specified host.

Format

host show [host_name]

Usage

The host show operation returns a list describing all processes that are configured in the server configuration data on the specified host. The optional argument is the fully qualified or cell-relative DCE name of a host. The local host is assumed if a host name is not given. The returned list is in the form:

- Server name as output by server catalog -simplename
- One of the tokens: running or not running
- An optional server-specific comment such as master or replica for a security server, and clerk or server for a DTS server.

Privilege Required: You must have r (read) permission to the config/srvrconf container object on the specified host.

Examples

The following example shows configured servers on host hydra:

dcecp> host show hosts/hydra
{myserver1 running}
{myserver2 not running}
dcecp>

Related Information

Commands:

account       dcecp       dts       server
aud           directory    registry
**host start**

Starts all DCE processes listed in the server configuration data on the specified host.

**Format**

```
host start [host_name]
```

**Usage**

The `host start` operation starts all DCE processes on the specified host. This command depends on the `dced` running on the specified host. For example, it cannot be used to start DCE on an MVS/ESA OpenEdition DCE Release 1 machine. The processes that are started are all those listed in the server configuration data stored in the `dced` on the specified host with the `boot` or `explicit` values of the `starton` attribute. You can add servers to the server configuration data by using the `server create` command.

The argument is the fully-qualified or cell-relative name of the host to operate on. If called with no argument, `dced` is started on the local host that requires the appropriate local permissions first, usually `root`. If a host name is specified, `dced` must be running on that host. Before starting any host, make sure that a security server and a CDS server are both running somewhere in the cell.

**Privilege Required:** You must have `x (execute)` permission to the `config/srvrconf` container object on the specified host.

**Examples**

The following example starts all DCE processes on host `hydra`:

```
dcecp> host start hosts/hydra
dcecp>
```

**Related Information**

Commands:

- `account`
- `dcecp`
- `dts`
- `registry`
- `server`
- `directory`
host stop

Stops all DCE processes listed in the server execution data on the specified host.

Format

host stop [host_name] [-force]

Options

-force  Optionally specifies that any servers that fail to stop normally are stopped using a server stop -method hard command.

Usage

The host stop operation stops running DCE processes on the specified host. This command depends on the dced running on the specified host. For example, it cannot be used to stop DCE on an MVS/ESA OpenEdition DCE Release 1 machine. The argument is the fully-qualified or cell-relative name of the host to operate on.

Processes are stopped as follows:

- All servers listed in the server execution data are stopped. Servers implementing DCE core services are stopped last, in an appropriate order.
- If any servers fail to stop, and the -force option was specified, those servers are stopped by a server stop -method hard command.

Privilege Required: You must have s (stop) permission on the config/srvexec object for each server being stopped.

Examples

The following example stops host hydra:

dcecp> host stop hosts/hydra
dcecp>

Related Information

Commands:

account  dcecp  dts  server
aud  directory  registry
hostdata

A dcecp object that manages the DCE host principal name and cell affiliation information on the host.

Format

```
hostdata catalog [host_name_list] [-simplename]
hostdata create hostdata_name_list [-attribute attribute_list | attribute options] [-entry] [-binary]
hostdata delete hostdata_name_list [-entry]
hostdata help [operation | -verbose] [-syntax]
hostdata modify hostdata_name_list [-change attribute_list | attribute options] [-binary]
hostdata operations
hostdata show hostdata_name_list [-entry] [-binary]
```

Parameters

- `host_name_list` A list of one or more names of hosts. Usually they are in the form:
  ```
  /.:/hosts/hostname
  ```
- `hostdata_name_list` A list of one or more names of host data items. Usually, they are in the form:
  ```
  /.:/hosts/hostname/config/hostdata/name
  ```
  although, a shortened notation referring to the local machine consisting of only name can be used as well.
- `operation` The name of one specific hostdata operation (subcommand) about which you want help information.

Usage

The hostdata object represents a host data entry stored by dced on a host that represents some data, usually a file. The data itself is represented by the hostdata/data attribute of the entry. Remote manipulation of this data is accomplished by these commands. The names of these host data objects are in the DCE name space and are served by dced.

Attributes

- `annotation comment` A human readable comment field limited to Portable Character Set (PCS) data that cannot be modified after creation.
- `hostdata/data string` An attribute that represents the actual data. Its syntax is a list of strings. The data can be viewed and modified in two different modes; as a string or as binary data. By default, the string mode is used, but some of the commands accept a binary option to allow this attribute to be displayed or modified in binary form. When viewed as a string, each string in the list represents one line in the host data file.
- `storage string` A PCS string that identifies the name of the data repository. In dced, it is a file name. It is required and cannot be modified after creation.
**uuid hexadecimal number**

An internal identifier for the host data entry. Its value is a UUID. If not specified on creation, one is generated by `dcecp`. **uuid** cannot be modified after creation.

See the [z/OS DCE Administration Guide](#) for more information about attributes.

**Related Information**

Commands:

- `dcecp`  
  `server`
hostdata catalog

Returns a list of the names of all host data entries on the specified host in arbitrary order.

Format

`hostdata catalog [host_name_list] [-simplename]`

Options

- `-simplename` Returns a list of host data entries that are relative, not fully qualified names.

Usage

The `hostdata catalog` command returns a list of the names of all host data entries on the specified host in arbitrary order. The argument is a list of host names. If more than one host name is specified, the output is concatenated. If no argument is given, the local host is assumed.

Privilege Required: You must have r (read) permission to the `/./hosts/hostname/config/hostdata` container on the host.

Examples

dcecp> hostdata catalog
/../mycell.goodco.com/hosts/mars/config/hostdata/dce_cf.db
/../mycell.goodco.com/hosts/mars/config/hostdata/cell_name
/../mycell.goodco.com/hosts/mars/config/hostdata/pe_site
/../mycell.goodco.com/hosts/mars/config/hostdata/host_name
/../mycell.goodco.com/hosts/mars/config/hostdata/cell_aliases
/../mycell.goodco.com/hosts/mars/config/hostdata/post_processors

Related Information

Commands:

dcecp server
hostdata create

Creates a host data configuration entry.

Format

hostdata create  hostdata_name_list  [-attribute  attribute_list \ attribute_options]  [-entry]  [-binary]

Options

-attribute  attribute_list

Lets you specify attributes using an attribute list rather than using the -uuid, -annotation, -storage, and -data options. The format is:

-attribute  {uuid  hex_number}  {annotation  comment}  {storage  PCSstring}
{data  string}

The -attribute option is intended for use in scripts when pasting in lengthy attribute lists output by previous commands. The individual attribute options may be easier to use for interactive commands.

-binary

Specifies that the value of the data attribute is in binary form.

-entry

Creates a hostdata entry using already existing data. The data file specified in the -storage option or storage attribute must already exist. The data for the contents of the file become the data for the hostdata entry. This option cannot be specified with the data option or data attribute.

The alternative attribute options are:

-annotation  comment

A human readable comment field limited to PCS data. This cannot be modified after creation.

-data  string

An attribute that represents the actual data. Its syntax is a list of strings. In the common case, each string represents one line in the host data file. This option cannot be specified when -entry is specified. When using the data option or data attribute, use only one hostdata name.

-storage  string

A PCS string that identifies the name of the data repository. In dced, it is a file name. It cannot be modified after creation. This option must be specified if attribute options are used. The -storage option requires either the -data or the -entry option.

-uuid  hexadecimal_number

An internal identifier for the host data entry. Its value is a UUID. If not specified on creation, one is generated by dcecp. This cannot be modified after creation. If more than one host data entry is being created, the specified UUID is used for the first host data entry and a new UUID is generated for each of the other host data entries.

Usage

The hostdata create command creates a host data configuration entry. hostdata create uses an -attribute option or attribute options to specify configuration information for dced. The contents of the host data file can be specified through the -data option or the data attribute. Specify the -entry option to create one or more host data entries using an already existing host data file. An empty string is returned upon successful completion.
hostdata create

Privilege Required: You must have i (insert) permission to the $/hosts/hostname/config/hostdata$ container on the host.

Examples

```
dcecp> hostdata create file1 -storage /tmp/file1 -data {{first line}}
dcecp> hostdata show file1
    {uuid 8484188a-eb85-11cd-91b1-9zerodot8RzerodotRzerodotRzerodot9251352}
    {annotation {}}
    {storage /tmp/file1}
    {hostdata/data {first line}}
dcecp> cat /tmp/file1
    first line
```

Related Information

Commands:

```
dcecp server
```
**hostdata delete**

Deletes a host data entry and its data.

**Format**

```plaintext
hostdata delete hostdata_name_list [-entry]
```

**Options**

- `-entry` Specifies that only the configuration information that dced keeps is deleted, not the actual host data. Specify this option if the host data is used by another hostdata entry or if a new hostdata entry will be created using the host data.

**Usage**

The `hostdata delete` command deletes a host data entry and its data. The argument is a list of names of host data entries being deleted in the order specified. If the `-entry` option is present, only the configuration information that dced keeps is deleted, not the actual host data. This allows the host data to be used by another host data entry. An empty string is returned upon successful completion.

**Privilege Required:** You must have dp (delete, purge) permission to the host data object, `/:/hosts/hostname/config/hostdata/name`. P permission is necessary to delete the actual host data.

**Examples**

```plaintext
dcecp> hostdata delete file1
dcecp>
```

**Related Information**

Commands:

```
dcecp
server
```
hostdata help

hostdata help

Returns help information about the hostdata object and its operations.

Format

hostdata help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each hostdata operation.
-verbose Displays information about the hostdata object.

Usage

The hostdata help command is used without an argument or option to return brief information about each hostdata operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the hostdata object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the hostdata help command.

Examples

dcecp> hostdata help
   catalog Returns the list of hostdata object names.
   create Creates a new hostdata configuration object.
   delete Deletes a hostdata object and its associated data.
   modify Modifies the data of a hostdata object.
   show Returns the attributes of a hostdata object.
   help Prints a summary of command-line options.
   operations Returns the valid operations for command.

dcecp>

Related Information

Commands:

dcecp server
hostdata modify

Changes attributes of a hostdata entry, including the host data itself.

Format

    hostdata modify hostdata_name_list {-change attribute_list | attribute options} [-binary]

Options

- **-change attribute_list**
  Specifies an attribute list of new values. Only the data attribute is valid.

- **-binary**
  Specifies that the value of the data attribute is in binary form.

The alternative attribute option is:

- **-data**
  Specifies the new value of the data attribute.

Usage

The hostdata modify operation changes attributes of a host data entry, including the host data itself. The argument is a list of names of host data entries being modified. If more than one is specified, all modifications specified are done to each host data entry listed. Only the data attribute is modifiable, and it can only be completely replaced. Accepts a -change option to specify an attribute list of one new value for the data attribute. Also accepts a -data option whose value is the new value of the data attribute.

**Privilege Required:** You must have w (write) permission to the host data object, /./hosts/hostname/config/hostdata/name.

Examples

    dcecp> hostdata mod file1 -data {{new first line}}
    dcecp> hostdata show file1  
    {uuid cda3a184-eb85-11cd-91b1-080009251352}
    {annotation {}}
    {storage /tmp/file1}
    {hostdata/data {new first line}}
    dcecp> cat /tmp/file1
    new first line
    dcecp>

Related Information

Commands:

    dcecp
    server
hostdata operations

Returns a list of the operations supported by the hostdata object.

Format

hostdata operations

Usage

The hostdata operations command uses no arguments, and returns a list of the available operations for the hostdata object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the hostdata operations command.

Examples

dcecp> hostdata operations
catalog create delete modify show help operations
dcecp>

Related Information

Commands:

dcecp server
hostdata show

Returns an attribute list of the host data entries specified in the argument.

Format

hostdata show hostdata_name_list [-entry] [-binary]

Options

-entry Only the configuration information that dced keeps is displayed, not the actual host data.

-binary Specifies to return the value of the data attribute in binary form.

Usage

The argument is a list of names of host data entries. If called with the -entry option, then the data attribute is not returned. A -binary option can be specified to return the value of the data attribute in binary form. If the argument is a list, the output is concatenated into a single list in the order specified.

Privilege Required: You must have r (read) permission to the host data object, /.:/hosts/hostname/config/hostdata/name.

Examples

dcecp> hostdata show /.:/hosts/mars/config/hostdata/cell_name
{uuid 00174f6c-6eca-1d6a-bf90-0000c09ce054}
{annotation {Name of cell}}
{storage cell_name}
{hostdata/data /.../mycell.goodco.com}
dcecp>

Related Information

Commands:

dcecp server
**keytab**

A dcecp object that manages server passwords on DCE hosts.

**Format**

```bash
keytab add keytab_name_list -member principal_name_list
{-random -registry [-version key_version] | -key plain_key -version key_version [-registry]}
[-local] [-noprivacy]
keytab catalog [host_name_list] [-simplename] [-local] [-noprivacy]
keytab create keytab_name_list {-attribute attribute_list | attribute options} [-random] [-entry] [-local]
[-noprivacy]
keytab delete keytab_name_list [-entry] [-local] [-noprivacy]
keytab help [operation | -verbose] [-syntax]
keytab list keytab_name_list [-local] [-noprivacy]
keytab operations
keytab remove keytab_name_list -member principal_name_list [-version key_version_list]
[-type key_type] [-local] [-noprivacy]
keytab show keytab_name_list [-entry | -members] [-keys] [-local] [-noprivacy]
```

**Parameters**

- **host_name_list** A list of one or more names of hosts. Usually they are in the form:
  ```bash
  /.:/hosts/host_name.
  ```

- **keytab_name_list** A list of one or more names of key tables to operate on. Key table names are similar to other dced objects with the form:
  ```bash
  /.../cellname/hosts/hostname/config/keytab/name
  ```

- **operation** The name of one specific keytab operation (subcommand) about which you want help information.

**Usage**

The keytab object represents key tables (usually files) that store server keys (and key version numbers) on hosts. These key tables are manipulated remotely using dced, which maintains configuration information about each key table. This information is represented as the attributes on this object. The keys are considered members of the key table container. The names of keytabs are similar to other dced objects, namely: `/.../cellname/hosts/hostname/config/keytab/name`.

A key table has a set of keys. Each key contains a principal name, type, version, and value. The value can be created and changed. Removal of a key is based on the name, type, and version number. The dcecp syntax of a key is a list of principal name, type (which is either plain or des), version (a non-negative integer), and value. The value of a des key is 64 bits long and can be represented in dcecp as extended registry attributes (ERAs) of type byte (refer to the xattrschema object attributes for details). The value is valid on input but is not displayed on output so that keys are not shown on the screen. For example:

```bash
melman des 1 <key1>
melman plain 3 <key2>
```
Multiple keys for the same principal are displayed as separate keys. See the example in the `keytab show` command.

**Attributes**

**uuid** *value*  
An internal identifier of the configuration information kept by `dced` for this key table. Its value is a UUID. If not specified on creation, one is generated. This cannot be modified after creation.

**annotation** *comment*  
A human readable comment field limited to Portable Character Set (PCS) data. This cannot be modified after creation.

**storage** *string*  
The name of the key table (usually a file name). It is required and cannot be modified after creation.

**data** *key_list*  
The contents of the key table. Represented as a list of keys.

See the [z/OS DCE Administration Guide](#) for more information about attributes.

**Related Information**

Commands:

- `dcecp`
- `xattrschema`
keytab add

keytab add

Adds members to a key table.

Format

```
keytab add keytab_name_list -member principal_name_list
[-random -registry [-version key_version] | -key plain_key -version key_version [-registry]]
[-local] [-noprivacy]
```

Options

- **-key plain_key** Specifies a key explicitly. Cannot be used with the -random option.
- **-local** Specifies to operate on the local key file only.
- **-member principal_name_list** Lists principal names being added to each key table in the argument. This option must be specified.
- **-noprivacy** Specifies not to encrypt key tables sent over the network.
- **-random** Generates a random des key. Cannot be used with the -key option.
- **-registry** Updates the principal key in the registry as well as on the host. Required if the -random option is used.
- **-version key_version** Specifies a version number for the key. Required if the -key option is used.

Usage

The keytab add command adds members to key tables. The argument is a list of the names of key tables to have members added. The value of the required -member option is a list of principal names being added to each key table in the argument. The principals must exist or the command returns an error. The principal is added to the key table along with a key. Use either -random to have dcecp generate a random des key, or use -key with a value to specify plain key explicitly. The same key (whether specified or randomly generated) is used on all principals being added to all key tables. The -registry option updates the principal key in the registry as well. It is required if -random is used. Before using the -registry option, a password for the principal must exist in the key tables specified in keytab_name_list. The -version option specifies the version number of the key. You must specify -registry, or -version, or both on any keytab add command. An empty string is returned upon successful completion.

Privilege Required: You must have a (append) permission to the keytab object, 
/.../cellname/hosts/hostname/config/keytab/name

Examples

The following example adds members to a key table.

```
dcecp> keytab add /./hosts/medusa/config/keytab/radiology \
> -member melman -random -registry
```

```
dcecp> keytab add /./hosts/medusa/config/keytab/radiology \
> -member melman -version 2 -key yrrebnesor
```

232  Command Reference
Related Information

Commands:

dcecp

xattrschema
keytab catalog

Returns a list of the names of all key tables on the host specified in the argument.

Format
keytab catalog [host_name_list] [-simplename] [-local] [-noprivacy]

Options
- **-local**
  Specifies to operate on the local key file only.
- **-noprivacy**
  Specifies not to encrypt key tables sent over the network.
- **-simplename**
  Returns key table names that are relative, not fully qualified, and the name of the keytab container.

Usage
The keytab catalog operation returns a list of the names of all key tables on the host specified in the argument. If a host name is not specified, the current host is used. If the argument is a list, the output is concatenated. The return order is arbitrary.

Privilege Required: You must have r (read) permission to the /.../cellname/hosts/hostname/config/keytab container on the host.

Examples
```sh
dcecp> keytab catalog
/.../mycell.goodco.com/hosts/medusa/config/keytab/self
dcecp>
```

Related Information
Commands:

```sh
dcecp xattrschema
dcecp
```
keytab create

Creates a key table.

Format

keytab create keytab_name_list {-attribute attribute_list \ attribute options} [-random] [-entry] [-local] [-noprivacy]

Options

-attribute attribute_list

Lets you specify attributes using an attribute list rather than using the -uuid, -annotation, -storage, and -data options. The format is:

-attribute {uuid hex number} {annotation comment} {storage PCSstring} {data string}

The -attribute option is intended for use in scripts when pasting in lengthy attribute lists output by previous commands. The individual attribute options may be easier to use for interactive commands.

-entry

Creates a keytab entry using already existing data. The data file specified in the -storage option or storage attribute must already exist. The data for the contents of the file become the data for the keytab entry. This option cannot be specified with the data option or data attribute.

-local

Specifies to operate on the local key file only.

-noprivacy

Specifies not to encrypt key tables sent over the network.

-random

Causes the command to generate a random key.

The alternative attribute options are:

-annotation comment

A human readable comment field limited to PCS data. This cannot be modified after creation.

-data key_list

Specifies the contents of the key table. Represented as a list of keys. This option cannot be used with -entry. You must specify data or -entry, but not both. When using the -data option or data attribute, specify only one keytab name.

-storage string

Specifies the name of the key table (usually a file name). It cannot be modified after creation. This option must be specified if attribute options are used. The -storage option requires either the -data or the -entry option.

-uuid value

Specifies an internal identifier of the configuration information kept by dced for this key table. Its value is a UUID. If not specified on creation, one is generated by dcep. This cannot be modified after creation. If more than one key table is being created, the specified UUID is used for the first key table and a new UUID is generated for each of the other key tables.
keytab create

Usage

The **keytab create** operation creates a key table. The argument is the name of the key table to create and uses an **-attribute** option to specify configuration information for **dced**. The contents of the key table can be specified using the **-data** option or the **data** attribute. The value of the option is applied to all elements of the argument list. An empty string is returned upon successful completion.

If specified, the value of the **-data** option or **data** attribute is a list of keys. Each key must have a principal name and key type. The version is optional. If not present, the system generates a version of 1. If the key type is **plain**, a key value must be specified. If the type is **des**, it may be specified. If not present, one is generated randomly only if the **-random** option is provided. Specify the **-entry** option to create one or more key tables using an already existing keytab file.

**Note:** The keytab file created by this command using the **-storage** attribute is owned by a root user ID in read/write mode. In z/OS, the root user ID STC1 is used. The keytab file has very restrictive access. It is the administrator's responsibility to change permissions, group, and ownership as necessary to enable principals to start the server.

To get access to this keytab file when a server is started:

- The server must have root authority, or
- The permissions of the file must be changed to give access to the server's local identity. For example, if the server is going to run from the GSERV1 user ID, the keytab file should be updated so that its owner is GSERV1.

**Privilege Required:** You must have **i** (insert) permission to the 
/.../cellname/hosts/hostname/config/keytab container on the host.

Examples

The following example creates two keys for user **melman** and one key for **pwang** on host **medusa**. One of **melman**'s keys is an automatically generated **des** (data encryption standard) key. **Melman**'s second key and **pwang**'s key are manually entered keys.

```
dcecp> keytab create /.:hosts/medusa/config/keytab/radiology -attr \
> {{storage /opt/dcelocal/keys/radiology} {data {{melman des} 
> {melman plain 3 <key2>} {pwang des 2 <key3>}}}}
dcecp>
```

**Related Information**

Commands:

```
dcecp                xattrschema
```
keytab delete

Deletes a key table entry and its data.

Format

```
keytab delete keytab_name_list [-entry] [-local] [-noprivacy]
```

Options

- **-entry**
  Specifies that only the configuration information that dced keeps is deleted, not the actual keytab data. Specify this option if the keytab data is used by another key table or if a new key table will be created using the keytab data.

- **-local**
  Specifies to operate on the local key file only.

- **-noprivacy**
  Specifies not to encrypt key tables sent over the network.

Usage

The `keytab delete` operation deletes a key table entry and its data. The argument is a list of names of key table entries being deleted in the order specified. If the `-entry` option is present, only the configuration information that dced keeps is deleted, not the actual key table. Returns an empty string on success.

Privilege Required: You must have d (delete) permission to the keytab object, `./.../cellname/hosts/hostname/config/keytab/name`. If you are removing the key table, you must have p (purge) permission to the keytab object as well.

Examples

```
dcecp> keytab delete /./hosts/medusa/config/keytab/radiology
dcecp>
```

Related Information

Commands:

```
dcecp xattrschema
```
keytab help

keytab help

Returns help information about the keytab object and its operations.

Format

keytab help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each keytab operation.
-verbose Displays information about the keytab object.

Usage

The keytab help command is used without an argument or option to return brief information about each keytab operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the keytab object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the keytab help command.

Examples

dcecp> keytab help
add Adds keys into a key table.
catalog Returns the list of key table names.
create Creates a new key table entry and its keys.
delete Deletes a key table and its associated data.
list Lists all principals in a specified key table.
remove Removes keys from a key table.
show Returns the list of keys of a key table.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp xattrschema
keytab list

Returns a list of all the principals in the specified key table.

Format

keytab list keytab_name_list [-local] [-noprivacy]

Options

-local Specifies to operate on the local key file only.
-noprivacy Specifies not to encrypt key tables sent over the network.

Usage

The keytab list operation returns a list of all the principals in the specified key table. If the argument is a list of key table names, the output is concatenated with a blank line between key tables.

Privilege Required: You must have r (read) permission to the keytab object on the host, /.../cellname/hosts/hostname/config/keytab/name

Examples

dcecp> keytab list /.:/hosts/medusa/config/keytab/self
/.../mycell.goodco.com/hosts/medusa/self
/.../mycell.goodco.com/hosts/medusa/cds-server
/.../mycell.goodco.com/hosts/medusa/cds-server

dcecp>

Related Information

Commands:

dcecp xattrschema
keytab operations

keytab operations

Returns a list of the operations supported by the keytab object.

Format

keytab operations

Usage

The keytab operations command uses no arguments, and returns a list of the available operations for the keytab object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the keytab operations command.

Examples

dcecp> keytab operations
add catalog create delete list remove show help operations
dcecp>

Related Information

Commands:

dcecp xattrschema
keytab remove

Removes a member from a key table.

Format

keytab remove keytab_name_list -member principal_name_list [-version key_version_list] [-type key_type] [-local] [-noprivacy]

Options

- **-local**  Specifies to operate on the local key file only.
- **-noprivacy**  Specifies not to encrypt key tables sent over the network.
- **-member principal_name_list**  Specifies a list of one or more principal names of members you want to remove from the key table.
- **-type key_type**  Specifies whether the key is a des (data encryption standard) key or a plain key.
- **-version key_version_list**  Specifies a version number for the key.

Usage

The keytab remove operation removes a member from a key tab. The argument is a list of names of key tables from which to remove members. The value of the required -member option is a list of names of principals being removed from the key tables listed in the argument. The -version and -type options can be used to limit the keys removed. If one or both of these options is present, then only keys matching the values of these options are removed. The value of the -version option can be a list of version numbers. An empty string is returned upon successful completion.

Privilege Required:  You must have e (expunge) permission to the keytab object on the host, /./cellname/hosts/hostname/config/keytab/name

Examples

The following example removes all des keys for principal D_Britt:

dcecp> keytab remove ./hosts/medusa/config/keytab/radiology -member D_Britt -type des
dcecp>

Related Information

Commands:

dcecp  xattrschema
keytab show

keytab show

Returns an attribute list of the key table entries specified in the argument.

Format

keytab show keytab_name_list [-entry | -members] [-keys] [-local] [-noprivacy]

Options

- **-entry** Returns the dced entry attributes only and not the actual key table. Cannot be specified with the -members option.
- **-keys** Returns the data attributes (key table) with the actual values of the keys.
- **-local** Specifies to operate on the local key file only.
- **-members** Returns the data attributes only (key table), without the actual keys. Cannot be specified with the -entry option.
- **-noprivacy** Specifies not to encrypt key tables sent over the network.

Usage

The keytab show operation returns an attribute list of the key tables specified in the argument. The argument is a list of names of key tables. If used with -entry, the data attribute (key table) is not returned. If the optional -members option is given, only the value of the data attribute is returned (a list of keys). Keys are not normally output unless the -keys option is used. If the argument is a list, the output is concatenated with a blank line between key tables.

Privilege Required: You must have r (read) permission to the keytab object on the host, /.../cellname/hosts/hostname/config/keytab/name

Examples

The following example shows the key types and versions in the radiology key table on host medusa for member melman:

dcecp> keytab show ./:/hosts/medusa/config/keytab/self -members
{[/.../mycell.goodco.com/hosts/medusa/keytab/self des 1]
{[/.../mycell.goodco.com/hosts/medusa/keytab/self plain3]
{[/.../mycell.goodco.com/hosts/medusa/keytab/self des 2]
dcecp>

Related Information

Commands:

dcecp xattrschema
A **dcecp** object that manages a soft link in the DCE Cell Directory Service (CDS).

### Format

- **link create** `link_name_list [-to target_name [-timeout time_list] | -attribute attribute_list]
- **link delete** `link_name_list`
- **link help** `[operation | -verbose] [-syntax]
- **link modify** `link_name_list {[-add attribute_list] [-remove attribute_list] [-change attribute_list]}
- **link operations**
- **link show** `link_name_list [-schema]`

### Parameters

- **link_name_list** A list of one or more names of soft links being operated on.
- **operation** The name of one specific **link** operation (subcommand) about which you want help information.

### Usage

A **link** object represents a CDS soft link. A soft link in the CDS directory service contains an attribute with a name of the object that the soft link points to. The soft link contains several built-in attributes, but users may add their own attributes. Soft links can point to objects, directories, and other soft links.

### Attributes

The following are the CDS-defined attributes that may be present in CDS link objects:

- **CDS_CTS** Specifies the creation timestamp (CTS) of this soft link.
- **CDS_LinkTarget** Specifies the full name of the directory, object entry, or other soft link to which the soft link points.
- **CDS_LinkTimeout** Specifies a timeout value after which the soft link is either renewed or deleted. Its value is a list of two elements enclosed in braces: `{expiration_time extension_time}`:
  - The expiration-time. A date and time after which CDS checks for the soft link target and either extends or deletes the soft link. The value is specified as `yyyy-mm-dd-hh:mm:ss`, and portions of it can be the default.
  - The extension-time. A period of time by which to extend the soft link expiration time (if the server validated that the target still exists). The value is specified as `ddd-hh:mm:ss`, and portions of it can be defaulted.
- **CDS_UTS** Specifies the timestamp of the most recent update to an attribute of the soft link. The value of this attribute is a DTS-style timestamp. The attribute is written by the system and is read-only to users.

See the [z/OS DCE Administration Guide](https://www.ibm.com/support/knowledgecenter) for more information about attributes.
Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>clearinghouse</th>
<th>directory</th>
<th>object</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>dcecp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
link create

Creates a new soft link entry in the Cell Directory Service.

Format

```
link create link_name_list [-to target_name [-timeout time_list] | -attribute attribute_list]
```

Options

-attribute attribute_list

Lets you specify CDS attributes using an attribute list. See “Attributes” on page 243 for more information about attributes.

-timeout time_list

Specifies the expiration time and an extension period for all soft links named by the
```
link_name_list
```
argument. The time_list syntax is:

```
{expiration_time extension_time}
```

If you omit the -timeout option or specify a value that is not correct, the link is permanent and must be explicitly deleted.

-to target_name

This option lets you specify one directory name for the links to point to. If you do not use this option, you must specify the link target with the -attribute option.

Usage

The link create operation creates a new soft link entry in the Cell Directory Service. The required
```
link_name_list
```
argument is a list of one or more full CDS names of the soft links being created.

Privilege Required: You must have i (insert) permission to the directory in which you intend to create the soft link.

Examples

The following command creates a permanent soft link named
```
../sales/tokyo/price-server
```
that points to an object entry named
```
../sales/east/price-server
```
. The expiration value specifies that CDS checks that the destination name
```
../sales/east/price-server
```
still exists on June 25, 1999, at 12:00 p.m. If the destination name still exists, the soft link remains in effect another 90 days. After that, CDS checks that the destination name exists every 90 days.

```
dcecp> link create ../sales/tokyo/price-server -to \
> ../sales/east/price-server -timeout {1999-06-25-12:00:00 \n> 90-00:00:00}
dcecp>
```

You can enter the same information as the above example by using the -attribute option as follows:

```
dcecp> link create ../sales/tokyo/price-server -attribute {
> {CDS_LinkTarget ../sales/east/price-server}
> {CDS_LinkTimeout
>   {expiration 1999-06-25-12:00:00}
>   {extension 90-00:00:00}}
> }
dcecp>
```
link create

Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>clearinghouse</th>
<th>directory</th>
<th>object</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>dcecp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
link delete

Removes a link entry from the Cell Directory Service.

Format

`link delete link_name_list`

Usage

The `link delete` operation removes a link entry from the Cell Directory Service. This is usually done through a client application. The required `link_name_list` argument is a list of one or more full CDS names of the link entry being removed. An empty string is returned upon successful completion.

**Privilege Required:** You must have `d (delete)` permission to the link entry, or `A (Admin)` permission to the directory that stores the link entry.

Examples

The following command deletes the link `./sales/tokyo/price-server`:

```
dcecp> link delete /./sales/tokyo/price-server
```

Related Information

Commands:

- `cdscache`
- `cdscp`
- `clearinghouse`
- `directory`
- `dcecp`
- `object`
link help

Returns help information about the link object and its operations.

Format

link help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each link operation.
-verbose Displays information about the link object.

Usage

The link help command is used without an argument or option to return brief information about each link operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the link object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the link help command.

Examples

dcecp> link help
create Creates the named link.
delete Deletes the named link.
modify Adds, removes, or changes an attribute in the named link.
show Returns the attributes of a link.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

cdscache clearinghouse directory object
cdscp dcecp
link modify

Changes attributes in the specified soft links.

Format

\texttt{link\ modify\ link\_name\_list\ \{[-add\ attribute\_list]\ [-remove\ attribute\_list]\ [-change\ attribute\_list]\}}

Options

-\texttt{-add\ attribute\_list}\ Adds\ one\ or\ more\ new\ attributes\ to\ a\ soft\ link\ or\ adds\ values\ to\ existing\ attributes\ when\ values\ are\ not\ already\ present.\ Adds\ an\ attribute\ type\ with\ no\ value\ by\ specifying\ an\ attribute\ type\ with\ no\ value.\n
-\texttt{-remove\ attribute\_list}\ Removes\ an\ entire\ attribute\ or\ some\ attribute\ values\ from\ a\ soft\ link.\ The\ value\ of\ this\ option\ is\ an\ attribute\ list.\ If\ only\ the\ attribute\ type\ is\ specified\ after\ the\ option,\ the\ entire\ attribute\ is\ removed.\ If\ an\ attribute\ type\ and\ value\ are\ specified,\ then\ only\ that\ value\ is\ removed.\ Only\ the\ CDS\_LinkTimeout\ can\ be\ removed.\n
-\texttt{-change\ attribute\_list}\ Changes\ one\ attribute\ value\ to\ another\ for\ a\ soft\ link.\ The\ value\ of\ this\ option\ is\ an\ attribute\ list.\ Each\ attribute\ in\ the\ list\ has\ its\ existing\ value\ replaced\ by\ the\ new\ value\ given\ in\ the\ attribute\ list.\ For\ multi-valued\ attributes,\ all\ existing\ values\ are\ replaced\ by\ all\ the\ values\ listed\ for\ the\ attribute\ in\ the\ attribute\ list.\n
Usage

The \texttt{link\ modify} operation\ can\ be\ used\ to\ change\ two\ attributes\ of\ a\ soft\ link, \texttt{CDS\_LinkTarget} and \texttt{CDS\_LinkTimeout}. The\ argument\ is\ a\ list\ of\ names\ of\ soft\ links\ being\ operated\ on.\ This\ command\ uses\ the\ \texttt{-add}, \texttt{-remove}, and \texttt{-change} options\ to\ specify\ an\ attribute\ list\ to\ describe\ the\ changes.\ All\ the\ changes\ are\ performed\ in\ the\ order\ specified\ in\ the\ command\ on\ each\ soft\ link\ named\ in the\ argument.\ An\ empty\ string\ is\ returned\ upon\ successful\ completion.

\textbf{Privilege Required:} You\ must\ have \texttt{w (write)}\ permission\ to\ the\ \texttt{link} object.

Examples

The following\ example\ sets\ the\ link\ expiration\ time\ to 1997\ and\ the\ extension\ time\ to\ ten\ days\ and\ zero\ hours.

\begin{verbatim}
dcecp> link modify /.:/depts/emergency -change {
> {CDS_LinkTimeout}
> {expiration 1997}
> {extension +10-0:0:0}}
dcecp>
\end{verbatim}

Related Information

Command:

\begin{verbatim}
cdscache clearinghouse directory object
cdscp dcecp
\end{verbatim}
link operations

Returns a list of the operations supported by the link object.

Format

link operations

Usage

The link operations command uses no arguments and returns a list of the available operations for the link object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the link operations command.

Examples

dcecp> link operations
create delete modify show help operations
dcecp>

Related Information

Commands:
cdscache clearinghouse directory object
cdscp dcecp
link show

Displays attribute information associated with specified link entries.

Format

link show link_name_list [-schema]

Options

-schema Returns only the name and type for each link attribute.

Usage

The link show operation displays attribute information associated with specified link entries. The required link_name_list argument is a list of one or more full CDS names of the soft links you want to show. If more than one link is shown, the attributes of all the soft links are concatenated into one list. The order of the returned attributes is the alphabetic order of the object identifier of each attribute for each object.

Privilege Required: You must have r (read) permission to the link entry.

Examples

The following command lists all of the attributes and their values associated with the link /./depts/emergency:

dcecp> link show /./depts/emergency
{CDS_CTS 1996-07-11-17:47:59.755+00:0010.000/00-00-c0-8a-df-56}
{CDS_UTS 1996-07-11-17:52:44.698+00:0010.000/00-00-c0-8a-df-56}
{CDS_LinkTarget /.../mycell.acme_health.org/depts/radiology}
{CDS_LinkTimeout
{expiration 1997-07-11-00:00:00.000}
{extension +10-10:00:00.000I------}}
dcecp>

Related Information

Commands:
cds cache
cd scp
cd scp
clearing house
directory
object

Chapter 2. DCE Administration Commands 251
log

A dcecp object that manages serviceability routing and debug routing.

Format

log help [operation | -verbose] [-syntax]
log list server_list [-comp component_name_list]
log modify server_list
   -change [{serviceability_routing_specifications} [debug_routing_specifications]]
log operations
log show server_list [-debug]

Parameters

operation The name of one specific log operation (subcommand) about which you want to help information.
server_list A list of servers on which to operate. Each server can be specified as either an
   RPC_server_namespace_entry or a string_binding_to_service.

RPC_server_namespace_entry
   Specifies the name space entry of the target server.

string_binding_to_service
   An RPC string binding that describes the target server network location. The value has the form of an RPC string binding, without
   an object UUID. The binding information contains an RPC protocol, a network address, and an endpoint within brackets:
   rpc-prot-seq:network-addr[endpoint]
   or
   object_uuid@rpc-prot-seq:network-addr[endpoint]

Data Structures

debug_routing_specifications
   Specifies DCE debug routing instructions. For a complete description of the ways that you can specify routings for serviceability messages, see the information that follows in "Setting Serviceability Debug Message Levels" on page 254.

serviceability_routing_specifications
   Specifies DCE serviceability routing instructions. For a complete description of the ways that you can specify routings for serviceability messages, see the information that follows in "Specifying Routing for Serviceability Messages."

Specifying Routing for Serviceability Messages: A serviceability_routing_specification is a space-separated list of serviceability routing elements. No spaces are allowed within the specification of an individual routing element. Each routing element is a sub-string consisting of four fields containing PCS (Portable Character Set) data, as follows, shown in string syntax form:

severity:output_form: destination[: application-defined]

where:
severity

One of the message severity levels:

- FATAL
- ERROR
- WARNING
- NOTICE
- NOTICE_VERBOSE

output_form

Specifies how messages of the associated severity level should be processed, and must be one of the following:

- **BINFILE**: Write these messages as binary log entries
- **TEXTFILE**: Write these messages as human-readable text
- **FILE**: Equivalent to **TEXTFILE**
- **DISCARD**: Do not record these messages
- **STDOUT**: Write these messages as human-readable text to standard output
- **STDERR**: Write these messages as human-readable text to standard error

The **BINFILE**, **TEXTFILE**, and **FILE** output_form specifiers may be followed by a two-number specifier of the form:

```
.gens.count
```

where:

- **gens**: An integer that specifies the number of files, for example, generations, that should be kept
- **count**: An integer specifying how many entries, for example, messages, should be written to each file

The multiple files are named by appending a dot to the simple specified name, followed by the current number. When the number of entries in a file reaches the maximum specified by count, the file closes, the generation number increments, and the next file opens. When the maximum number of files are created and filled, the generation number is reset to 1, and a new file with that number is created and written to (thus overwriting the already-existing file with the same name), and so on. Thus, the files wrap around to their beginning, and the total number of log files never exceeds gens, although messages continue to be written as long as the program continues writing them.

destination

Specifies where the message is sent, and is a path name. You can leave this field if the output_form specified is **DISCARD**, **STDOUT**, or **STDERR**. The field can also contain a %ld string in the file name which, when the file is written, is replaced by the process ID of the program that wrote the messages. File names cannot contain colons (:), semicolons (;), percent signs (%), or spaces.

application-defined

Used for application-specific information. Standard DCE programs ignore this.

### String Syntax

The string syntax for a serviceability routing specification is:

```
severity:output_form:destination [:application-defined][;...]
```

You can define multiple routing specifications as a semicolon-separated list.

---

**log**

---

Chapter 2. DCE Administration Commands 253
For example, the following specification:

```
FATAL:TEXTFILE:/dev/console;STDOUT:
ERROR:TEXTFILE.5.100:/tmp/errors
EXIT:DISCARD:
*:FILE:/tmp/svc-log
NOTICE:BINFILE:/tmp/log%ld
WARNING:STDOUT:
```

instructs the serviceability mechanism to:

- Send fatal error messages to the console and to standard output
- Send other error messages to a log-rolled file
- Discard normal exit reports
- Write all messages to a log file
- Send informational messages to a temporary binary log
- Send warnings to standard output

### Tcl Syntax

The Tcl syntax for a serviceability routing specification uses the format:

```
{severity output_form destination application-defined}
```

For example, the specification examples shown in the string syntax format would be expressed in Tcl syntax as follows:

```
{FATAL { {TEXTFILE /dev/console} STDOUT} }
{ERROR TEXTFILE.5.100 /tmp/errors}
{EXIT DISCARD}
{*:FILE /tmp/svc-log}
{NOTICE BINFILE /tmp/log%ld }
{WARNING STDOUT {} }
```

### Setting Serviceability Debug Message Levels:

Nine serviceability debug message levels (specified respectively by single digits from 1 to 9) are available. The precise meaning of each level varies with the application or DCE component in question, but the general notion is that ascending to a higher level (for example, from 2 to 3) increases the level of informational detail in the messages.

Setting debug messaging at a certain level means that all levels up to and including the specified level are enabled. For example, if the debug level is set at 4, then the 1, 2, and 3 levels are enabled as well.

The general format for the debug level specifier string is: `component:sub_comp.level,sub_comp.level,...`

Where:

- **component** is the three-character serviceability component code for the program whose debug message levels are being specified
- **sub_comp.level** is a serviceability subcomponent name, followed (after a dot) by a debug level (expressed as a single digit from 1 to 9). Note that multiple subcomponent/level pairs can be specified in the string.

If there are multiple subcomponents and it is desired to set the debug level to be the same for all of them, then the form:

```
component:*:level
```
will do this (where the "*" specifies all subcomponents)

**Routing Serviceability Debug Messages:** The routing is specified by the contents of a specially-formatted string that is either included in the value of the environment variable or the contents of the routing file.

The general format for the debug routing specifier string is:

```
component:sub_comp.level,...:out_form:dest[;out_form:dest...]
[GOESTO:{sev|component}]
```

Where `component`, `sub_comp.level`, `out_form`, `dest`, and `sev` have the same meanings as defined earlier.

For example, consider the following string value:

This value, when assigned to the `SVC_HEL_DBG` environment variable, would set the debug level and routing for all `hel` subcomponents. A debug level of 4 is specified, and all debug messages of that level or lower will be written both to standard error, and in text form to the file `/tmp/hel_debug_log_process_ID`, where `process_ID` is the process ID of the program writing the messages.

**Usage**

The `dcecp log` object represents the current state of message routing for a given server. It supports routing for both serviceability and debug messages. Debug routing may be removed from production environment servers while still being used by application servers.

The `log list`, `log modify`, and `log show` commands specify a `log` object by identifying the server on which the `log` object resides. You can identify the target server by supplying either the server entry in the name space, or a fully bound string binding. You can specify multiple target servers as a space-separated list. When specifying multiple servers, you can mix the name space entry and string binding forms in the same list.

**Note:** Not all servers support the serviceability interface that is used by the `log` command. If you enter a `log` command to one of these servers, you may receive one of the following messages:

```
EUVA07172E DCE interface error.
  DCE status code: 0x16c9a01d - Binding is not correct.

EUVA07172E DCE interface error.
  DCE status code: 0x16c9a02c - Interface is not known.
```

**Related Information**

Commands:

- `dcecp`
log help

Returns help information on the log object.

**Format**

`log help [operation | -verbose] [-syntax]

**Options**

- `-syntax` Displays the syntax diagram of each log operation.
- `-verbose` Displays information about the log object.

**Usage**

The log help command is used without an argument or option to return brief information about each log operation.

Use the optional `operation` argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the `-verbose` option for more detailed information about the log object itself.

Use the `-syntax` option to display the syntax diagram of each operation or of only the operation specified using the `operation` argument.

**Privilege Required:** No special privileges are needed to use the log help command.

**Examples**

dcecp> log help  
list        Returns serviceability components registered by a server.
modify     Changes serviceability routing specifications of a server.
show       Returns serviceability routing settings for a server.
help       Prints a summary of command-line options.
operations Returns the valid operations for command.

dcecp>

**Related Information**

Commands:

dcecp
log list

Returns a list of serviceability components registered by the target server or servers.

Format

log list server_list [-comp component_name_list]

Options

-comp component_name_list
  Specifies a list of DCE serviceability component names.

Usage

If you specify more than one server, the returned lists for the second and subsequent servers are concatenated to the returned list for the first server.

The -comp option specifies a space-separated list of DCE serviceability component names. For each named component, the command returns a list of the associated subcomponents. For each subcomponent in the list, the command displays its name, its level, and its description. The order of the component names is arbitrary. If you specify more than one component name, the resulting subcomponent lists are concatenated.

Privilege Required: No special privileges are needed to use the log list command.

Examples

dcecp> log list ./:hosts/medusa/cds-server
svc cds dts rpc sec
dcecp> log list ./:hosts/medusa/cds-server -comp dts
general 0 "General server administration"
{events 0 "Events received and acted upon"}
{arith 0 "Math operations"}
{ctmsgs 0 "Control messages received"}
{msgs 0 "Messages received"}
{states 0 "Server state transitions"}
{threads 0 "Thread interactions"}
{config 0 "Server/cell configuration"}
{sync 0 "Server sync interactions"}
dcecp>

Related Information

Commands:

dcecp
log modify

Sets message routing specifications for one or more specified servers.

Format

log modify server_list -change [{serviceability_routing_specifications} [debug_routing_specifications]]

Options

-change serviceability_routing_specifications debug_routing_specifications
   Specifies the serviceability routing or debug routing specifications that you want to change.

Usage

There is a fixed, well-known set of routing defaults. You can change these defaults, but you cannot add new routings or remove existing routings. On successful completion, log modify returns an empty string.

Routing is always set on a per-server basis and is recorded in the log object for each server.

You can specify multiple target servers as a space-separated list. Specify each server by supplying either the RPC string binding that describes the server's network location (string_binding_to_server) or a name space entry of the server (RPC_server_namespace_entry). When specifying multiple servers, you can mix the forms in the same list. You can also specify multiple routing specifications (serviceability and debug) as a space-separated list.

Privilege Required: The privileges are determined by what the server allows for permissions.

Examples

```
dcecp> log modify ./:/tserver -change {{FATAL TEXT FILE /dev/console} \ 
   > {ERROR TEXTFILE /tmp/timop_errors.5.1wzerodotwzerodot} {NOTICE BINFILE /tmp/timop_log%ld }}
dcecp>
```

Related Information

Commands:

```dcecp```
log operations

Returns a list of the operations supported by the log object.

Format

log operations

Usage

The log operations command uses no arguments and returns a list of the operations supported by the log object. The order of the elements is alphabetic except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the log operations command.

Examples

dcecp> log operations
list modify show help operations
dcecp>

Related Information

Commands:

dcecp
log show

log show

Returns a list describing the current serviceability routing settings for a server.

Format

log show server_list [-debug]

Options

-debug Specifies to return debug routing settings, rather than serviceability routing settings.

Usage

If you specify more than one server, the returned routings for the second and subsequent servers are concatenated to the returned routings for the first server. The order of the returned routing settings is arbitrary.

The -debug option instructs log show to return debug routing settings rather than serviceability routing settings. It is not available on servers for which debugging was turned off (for example, production servers).

Privilege Required: No special privileges are needed to use the log show command.

Examples

dcecp> log show ./:hosts/acme/cds-clerk
{ERROR STDERR -}
{FATAL FILE /dev/console}
{WARNING FILE /tmp/warnings.log}
dcecp>

Related Information

Commands:

dcecp
name

A `dcecp` object that compares and expands DCE names.

**Format**

```
name compare name name
name expand name
name get prot_seq:IP_addr
name help [operation | -verbose] [-syntax]
name operations
name parse name
```

**Parameters**

- `IP_addr` Specifies the IP address of the host name.
- `name` A name of an object in the DCE name space. Examples of names include principal names, names of security groups, names of CDS objects like directories, soft links, child pointers and so on, names of `rpcentries` and `rpcgroups`, and DFS file names.
- `operation` The name of one specific `name` operation (subcommand) about which you want help information.
- `prot_seq:IP_addr` Specifies the protocol sequence and IP address of the host name.

**Usage**

The `name` command resolves, compares, and parses DCE names.

**Related Information**

Commands:

- `dcecp`
name compare

name compare

Compares two names.

Format

name compare name name

Usage

The name compare command compares two names given as arguments and returns 1 if they both syntactically refer to the same name. Otherwise, returns 0.

Privilege Required: No special privileges are needed to use the name compare command.

Examples

dcecp> name compare /.:hosts /.:/subsys
0
dcecp>

dcecp> name compare /.../mycell.goodco.com/hosts /.:/hosts
1
dcecp>

Related Information

Commands:

dcecp
name expand

Expands a simple DCE name to a global name.

Format

name expand name

Usage

The name expand command uses a single name as an argument and returns the canonical form of the name. This has the affect of converting .:/ to ./cellname.

Privilege Required: No special privileges are needed to use the name expand command.

Examples

dcecp> name expand .:/cell-profile

/.../mycell.goodco.com/cell-profile

dcecp>

Related Information

Commands:

dcecp
name get

name get

Gets a host name from a binding.

Format

name get prot_seq:IP_addr

Usage

The name get command is used with a specified protocol sequence and an IP address to return the machine name.

Privilege Required: No special privileges are needed to use the name get command.

Examples

dcecp> name get ncacn_ip_tcp:129.35.20.1.
hosts/medusa
dcecp>

Related Information

Commands:

dcecp
name help

Returns help information about the name object and its operations.

Format

name help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each name operation.
-verbose Displays information about the name object.

Usage

The name help command is used without an argument or option to return brief information about each name operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the name object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the name help command.

Examples

dcecp> name help
compare Compares two names syntactically.
expand Returns the canonical form of a name.
get Gets host name from a partial or full string binding.
parse Parses name into cell name and residual name.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp
name operations

Returns a list of the operations supported by the name object.

Format

name operations

Usage

The name operations command uses no arguments, and returns a list of the available operations for the name object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the name operations command.

Examples

dcecp> name operations
compare expand get parse help operations
dcecp>

Related Information

Commands:

dcecp
name parse

Divides a name into a cell name and a residual name.

Format

name parse name

Usage

The name parse command parses a name into a cell name and a residual name. The argument is a single DCE name. Returns a list of two elements, cell name and residual name. A name not beginning with a slash (/) is considered a name in the local cell.

Privilege Required: No special privileges are needed to use the name parse command.

Examples

dcecp> name parse /.:/hosts
/.../mycell.goodco.com hosts
dcecp>

Related Information

Commands:

dcecp
object

A dcecp object that manages an object in the name service.

Format

object create object_name_list [-attribute attribute_list [-single]]
object delete object_name_list
object help [operation | -verbose] [-syntax]
object modify object_name_list
([-add attribute_list [-single]] [-remove attribute_list [-types]] [-change attribute_list])

object operations

object show object_name_list [-schema]

Parameters

object_name_list A list of one or more names of CDS objects. Examples of objects are RPC entries,
group entries, and profile entries.

operation The name of one specific object operation (subcommand) about which you want help information.

Usage

An object object represents a CDS object that identifies a resource such as a host system, a printer, an
application, or a file. An object in the DCE Directory Service has a CDS name and associated attributes
that are stored in CDS. Attributes consist of a type and one or more values. Every object is the child of a
CDS directory.

Attributes

The following are the CDS-defined attributes that may be present in CDS objects:

CDS_Class Specifies the class to which an object belongs.
CDS_CTS Specifies the creation timestamp (CTS) of this CDS object.
CDS_ClassVersion Contains the version number of the object class. This allows applications to build
compatibly with entries created by earlier versions.
CDS_ObjectUUID Specifies the unique identifier of the object being referenced.
CDS_UTS Specifies the timestamp of the most recent update to an attribute of the object entry.

You can also specify application-defined attribute names and values using the form:
{attribute_name value}

attribute_name An attribute name is specified as a simple name limited to 31 characters.
value A value is specified as a string and is limited to 31 characters.

See the z/OS DCE Administration Guide for more information about attributes.
Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>clearinghouse</th>
<th>directory</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>dcecp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
object create

Creates a new object entry in the Cell Directory Service.

Format

object create object_name_list [-attribute attribute_list [-single]]

Options

-attribute attribute_list
  Lets you specify CDS attributes using an attribute list (see "Attributes on page 268").

-single
  Valid only with the -attribute option, this specifies that attribute values are single-valued. Otherwise, attributes are multi-valued.

Usage

The object create operation creates a new object entry in the Cell Directory Service. This task is usually done through a client application. The required object_name argument is a list of the full CDS names of the object entries being created.

Optionally, you can use the -attribute option to associate one or more attributes (see "Attributes on page 268") with each object being created. The attribute values are multi-valued unless the -single option is specified, in which case all attributes are single-valued. The -single option is only valid if the -attribute option is specified.

Privilege Required: You must have i (insert) permission to the parent directory.

Examples

The following command creates an object entry named /.:sales/east/floor1cp. The object entry describes a color printer on the first floor of the Eastern sales office of a company.

dcecp> object create /.:sales/east/floor1cp -attribute \
> {{CDS_Class printer} {CDS_ClassVersion 1.0}}
dcecp>

Related Information

Commands:

cdscache clearinghouse directory link

cdscp dcecp
object delete

Removes an object entry from the Cell Directory Service.

Format

```
object delete object_name_list
```

Usage

The **object delete** operation removes an object entry from the Cell Directory Service. This is usually done through a client application. The required **object_name** argument is a list of the full CDS names of the object entries being removed.

**Privilege Required:** You must have **d (delete)** permission to the object entry, or **A (Admin)** permission to the directory that stores the object entry.

Examples

The following command deletes the object entry `./sales/east/floor1pr2`:

```
dcecp> object delete ./sales/east/floor1pr2
```

**Related Information**

Commands:

```
cdscache    clearinghouse    directory    link
cdscp       dcecp
```
object help

Returns help information about the object object and its operations.

Format

object help [operation [ -verbose] [-syntax]]

Options

-syntax Displays the syntax diagram of each object operation.
-verbose Displays information about the object object.

Usage

The object help command is used without an argument or option to return brief information about each object operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the object object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the object help command.

Examples

dcecp> object help
create Creates the named object.
delete Deletes the named object.
modify Adds, removes, or changes an attribute in the named object.
show Returns the attributes of an object.
help Prints a summary of command-line options.
operations Returns the valid operations for command.

dcecp>
**object modify**

Adds or removes attributes or changes attribute values for object entries in the Cell Directory Service.

**Format**

```
object modify object_name_list
([-add attribute_list [-single]] [-remove attribute_list [-types]] [-change attribute_list])
```

**Options**

- **-add attribute_list** Adds one or more new attributes to an object entry.
- **-change attribute_list** Changes one attribute value to another for an object entry. The value of this option is an attribute list. Each attribute in the list has its existing value replaced by the new value given in the attribute list. For multi-valued attributes, all existing values are replaced by all the values listed for the attribute in the attribute list.
- **-remove attribute_list** Eliminates one or more attribute values from an attribute type of an object entry. For instance, removing a value from an attribute with three values leaves the attribute with two values. The argument is an attribute list in the form:

  ```
  {{attribute_name attribute_value}...{attribute_name attribute_value}}
  ```

  To remove an attribute type as well as its values, use the **-types** option with the **-remove** option. For example,

  ```
  object modify /.:/foo -remove {RPC_CLASS RPC_CLASS_VERSION} -types
  ```

  If an attribute is not present, an error is returned. Fixed CDS attribute types such as the CDS creation Timestamp (CDS_CTS), cannot be removed.

  You cannot remove a single valued attribute like:

  ```
  dcecp -c directory modify /.:/DirName -remove {dirregion 6}
  ```

  You can only remove a multi-valued value, not the value of a single-valued attribute. However, if you want to remove the attribute altogether, use the **-types** option. This works for both multi and single valued attributes.

- **-single** May be used with the **-add** option to specify that the attribute is single-valued.
- **-types** Used with the **-remove** option to specify that the value of the **-remove** option is a list of attribute types. This means that the entire attribute is removed. This option is not valid without the **-remove** option.

**Usage**

The **object modify** operation adds, or removes attributes or changes attribute values for object entries in the Cell Directory Service. This task is usually done through a client application. The required **object_name_list** argument is a list of the full CDS names of the object entries being modified. If more than one of the **-add**, **-remove**, and **-change** options are specified, they are processed in the order specified in the command.

**Privilege Required:** You must have w (write) permission to the object entry.
object modify

Examples

To add the value of the sales_record attribute to region2 of an object entry named /.:Q1_records, follow these steps:

1. Read the cds_attributes file (/opt/dcelocal/etc/cds_attributes) to check that the attribute sales_record is listed, as shown in the following display:

<table>
<thead>
<tr>
<th>OID</th>
<th>LABEL</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.22.1.3.66</td>
<td>sales_record</td>
<td>char</td>
</tr>
</tbody>
</table>

2. Enter the following command to assign the value region2 to the attribute sales_record of an object entry named /.:Q1_records:

   dcecp> object modify /.:Q1_records -add {sales_record region2}

   dcecp>

Related Information

Commands:

cdscache  clearinghouse  directory  link
cdscp     dcecp
object operations

Returns a list of the operations supported by the object object.

Format

object operations

Usage

The object operations command uses no arguments, and returns a list of the available operations for the object object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the object operations command.

Examples

dcecp> object operations
create delete modify show help operations
dcecp>

Related Information

Commands:

cdscache clearinghouse directory link
cdscp dcecp
object show

Displays attribute information associated with specified object entries.

Format

object show object_name_list [-schema]

Options

-schema  This option returns whether an attribute is single or multi-valued. This is specific to an object, meaning that the same attribute can be single-valued on one object and multi-valued on another object.

Usage

The object show operation displays attribute information associated with specified object entries. The required object_name_list argument is a list of one or more full CDS names of the object entries being shown. If more than one object is shown, the attributes of all the objects are concatenated into one list. The order of the returned attributes is the alphabetic order of the object identifier of each attribute for each object.

The -schema option displays whether an attribute is single-valued or multi-valued.

Privilege Required: You must have r (read) permission to the object entry.

Examples

dcecp> object show ./:/obj
{RPC_ClassVersion
 {02 00}
 {03 00}}
{RPC_Group
 {12 34}}
{CDS_CTS 1996-07-01-22:06:54.990-05:0010.000/00-00-c0-f7-de-56}
{CDS_UTS 1996-07-01-22:07:37.248-05:0010.000/00-00-c0-f7-de-56}
{CDS_Class 02 00}
dcecp>

The following example illustrates use of the -schema option:

dcecp> object show ./:/obj -schema
{RPC_ClassVersion multi}
{RPC_Group multi}
{CDS_CTS single}
{CDS_UTS single}
{CDS_Class single}
dcecp>
Related Information

Commands:

<table>
<thead>
<tr>
<th>cdscache</th>
<th>clearinghouse</th>
<th>directory</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdscp</td>
<td>dcecp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
organization

A dcecp object that manages an organization in the DCE Security Service.

Format

- `organization add organization_name_list -member member_name_list`
- `organization catalog [server_name_list] [-simplename]`
- `organization create organization_name_list [-attribute attribute_list | attribute options]`
- `organization delete organization_name_list`
- `organization help [operation | -verbose] [-syntax]`
- `organization list organization_name_list [-simplename]`
- `organization modify organization_name_list [-change attribute_list | attribute options] | [-add extended_registry_attribute_list | -remove extended_registry_attribute_list [-types]]`
- `organization operations`
- `organization remove organization_name_list -member member_name_list`
- `organization rename organization_name -to new_organization_name`
- `organization show organization_name_list [-policies] [-xattrs] [-all]`

Parameters

- `operation` The name of one specific organization operation (subcommand) about which you want help information.
- `organization_name` The name of one security organization being renamed.
- `organization_name_list` A list of one or more names of organizations to act on. The names can all be fully qualified organization names as in:
  
  /.../cellname/organization_name
  
  Or, they can all be cell-relative organization names as in:
  
  organization_name

  Cell-relative names refer to an organization in the cell named in the _s(sec) convenience variable or in the default cell of the local host if the _s(sec) convenience variable is not set.

  Do not mix fully-qualified names and cell-relative names in a list. And, do not accidentally use names of registry database objects that contain registry information about organizations (organization names cannot begin with /./sec/org).

- `server_name_list` A list of one or more names of registry servers to act on. The names can be fully qualified as in:
  
  /.../cellname/subsys/dce/sec/registry_name
  
  Or, they can be cell-relative names as in:
  
  subsys/dce/sec/registry_name
Usage

The `organization` object represents registry organizations. Organizations are collections of principal names. Unless otherwise stated, all of the operations of this object use one argument, which is the name of the organization to act on. It must be an organization name, not the name of the registry database object that contains the registry information about the organization (that is, it begins with `/:sec/org/`).

After this command runs, the `_b(sec)` convenience variable is set to the name of the server that was bound to for the command. The value of the variable `_s(sec)` before the command is treated as a hint; the server specified is contacted if it can service the request. A case where it cannot service the request is if a read-only registry is bound to, and the next command is a `create` command. In this case, the master registry is bound to automatically and the `_b(sec)` variable is updated appropriately. The value of the variable is the name of the registry bound to in one of the formats specified as valid for the argument to the `registry` object.

Attributes

The `organization` object supports two kinds of attributes, organization and policy. Organization attributes may or may not have default values. They assume a default value or a value set by administrators. Policy attributes regulate such things as account and password lifetimes for all accounts associated with a particular organization. Policy attributes have registry-wide default values, set with the `registry modify` command. Policy attributes always assume the most restrictive value, whether it is the registry-wide default value or a value set by administrators.

The organization attributes are:

- `fullname string` Used with the `create` and `modify` operations for information purposes only. It usually describes or expands a primary name to allow easy recognition by users. For example, an organization can have a primary name of `osf` and a full name of `Open Software Foundation`. The value is a string if it contains spaces. It displays in quotation marks, and must be in quotation marks or braces (as per Tcl quoting rules) when entered. If not entered, the full name defaults to the null string (that is, blank).

- `orgid integer` Used with the `create` operation to specify the Organization Identifier for the organization. If this attribute is not present, then an `orgid` is assigned to the organization automatically.

- `uuid hexadecimal number` Used with the `create` operation as an internal identifier for an organization. No two organizations can have the same UUID. Usually, this attribute is set by the system and displays for administrators. In some unusual cases, such as adopting an orphaned UUID, an administrator might want to set this value explicitly.

The policy attributes are:

- `acctlife (relative_time | unlimited)` This policy attribute defines the life span of accounts. Specify the time using the DTS relative time format (`[-]dd-hh:mm:ss`) or the string `unlimited`.

- `pwdalpha {yes | no}` This policy attribute defines whether passwords can consist entirely of alphanumeric characters. Its value is either `yes` or `no`.

- `pwdexpdate date` This policy attribute defines a date when a password expires. Specify the date using an ISO compliant time format such as `(CC-MM-DD-hh:mm:ss)`, or the string `none`, which specifies that the password does not expire on a given date.
-pwdlife \{relative_time \| unlimited\}
This policy attribute defines the life span of passwords. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss) or the string unlimited.

-pwdminlen integer
This policy defines the minimum number of characters in a password. Its value is a positive integer or the integer 0, which means there is no minimum length.

-pwdspaces \{yes \| no\}
This policy attribute defines whether passwords can consist entirely of spaces. Its value is either yes or no.

See the [z/OS DCE Administration Guide](#) for more information about attributes.

**Related Information**

Commands:

- account
- dcecp
- group
- principal
- registry
- rgyedit
organization add

Adds members to a security organization.

Format

organization add organization_name_list -member member_name_list

Options

-member member_name_list
    A list of one or more names of principals to be added to each organization in the
    argument.

Usage

The organization add operation adds members to an organization. The argument is a list of names of
organizations having members added. The value of the required -member option is a list of names of
principals to be added to each organization in the argument. The principals must exist or the command
returns an error. An empty string is returned upon successful completion.

Privilege Required:  You must have rM (read, Member list) permissions on the target organization
and rg (read, groups) permissions on the principal being added.

Examples

dcecp> organization add managers -member W_Ward

dcecp>

Related Information

Commands:

account       group       registry
 dcecp         principal    rgyedit
organization catalog

Returns a list of the names of all organizations in the registry.

Format

organization catalog [server_name_list] [-simplename]

Options

-simplename Returns a list of organization names in the registry that are relative, not fully qualified.

Usage

The organization catalog operation returns a list of the names of all organizations in the registry database. This command uses an optional server name argument to specify the server to use. By default, fully qualified names are returned in the form cellname/organization_name. If the -simplename option is given, then the organization cell name is relative, not fully qualified.

Privilege Required: You must have r (read) permission to the /.:/sec/org directory.

Examples

dcecp> organization catalog
/.../mycell.goodco.com/none
/.../mycell.goodco.com/users
/.../mycell.goodco.com/managers

dcecp>

dcecp> organization catalog -simplename
none
users
managers

dcecp>

Related Information

Commands:

account group registry
dcecp principal rgyedit
organization create

Creates a new organization in the registry database.

Format

organization create organization_name_list {-attribute attribute_list \ attribute options}

Options

-attribute attribute_list

Lets you specify attributes using an attribute list rather than using the individual attribute options such as -orgid, -uuid, -acctlife, and -pwdalpha. The format is:

-attribute {{orgid integer} {uuid hex number} {acctlife value} {pwdalpha string}}

The -attribute option is intended for use in scripts when you can paste in lengthy attribute lists output by previous commands. The individual attribute options might be easier to use for interactive commands. Note that you can specify extended registry attributes (ERAs) with the -attribute option. You cannot specify ERAs as individual attribute options.

Alternative attribute options you can modify are:

-acctlife {relative_time | unlimited}

This policy attribute defines the life span of accounts. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss) or the string unlimited.

-fullname string

This attribute is used for information purposes only. It usually describes or expands a primary name to allow easy recognition by users. For example, an organization can have a primary name of osf and a full name of Open Software Foundation. The value is a string. If it contains spaces, it displays in quotation marks, and must be in quotation marks or braces (as per Tcl quoting rules) when entered. If not entered, the full name defaults to the null string (that is, blank).

-orgid integer

Specifies the Organization Identifier for the organization. If this attribute is not present, an orgid is automatically assigned to the organization.

-pwdalpha {yes | no}

This policy attribute defines whether passwords can consist entirely of alphanumerical characters. Its value is either yes or no.

-pwdexpdate date

This policy attribute defines a date when a password expires. Specify the time using an ISO compliant time format (such as CC-MM-DD-hh:mm:ss) or the string none, in which case, there is no expiration date for the password.

-pwdlife {relative_time | unlimited}

This policy attribute defines the life span of passwords. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss) or the string unlimited.

-pwddminlen integer

This policy attribute defines the minimum number of characters in a password. Its value is a positive integer or the integer 0, meaning there is no minimum length.

-pwddspaces {yes | no}

This policy attribute defines whether passwords can consist entirely of spaces. Its value is either yes or no.
organization create

-uuid hexadecimal number

This attribute is an internal identifier for an organization. No two organizations can have the same UUID. Usually, this attribute is set by the system and displays for administrators. In some unusual cases, as adopting an orphaned UUID, an administrator may want to set this value explicitly.

Usage

The organization create operation creates a new organization. The argument is a list of names of organizations being created. An empty string is returned upon successful completion. Options specify the attributes of the newly created organization. All options are applied to all organizations in the argument list.

Do not use the -orgid option when creating two or more organizations with the same command. The second create operation fails because the orgid is already in use after the first is created.

The create operation supports both attribute lists (with an -attribute option) and attribute options named after each attribute and policy. If an orgid is not entered, then one is assigned to the organization automatically. Only specify a UUID to adopt an orphaned UUID. Usually, the UUID for a new organization is generated by the registry. In cases where data exists tagged with a UUID of an organization that was deleted from the registry, it can be entered on the create command line to specify the old UUID for a new organization. The UUID specified must be an orphan (a UUID for which no name exists in the registry). An error occurs if you specify a name that is already defined in the registry. The orgid attribute may not be specified if the uuid is, but the fullname attribute may be.

Privilege Required: You must have i (insert) permission to the /.:/sec/org directory.

Examples

dcecp> organization create temps -fullname "Temporary Employees"
dcecp>

dcecp> organization create temps -attribute {fullname "Temporary Employees"}
dcecp>

dcecp> org create dce -fullname {Dist Comp Env} -orgid 101
dcecp>

dcecp> org create dce -fullname {Dist Comp Env} \ 
> -uuid c2aaac790-dc6c-11cc-a6f8-080009251352
dcecp>

Related Information

Commands:

account          group          registry
 dcecp          principal       rgyedit
organization delete

Deletes organizations from the registry.

Format

organization delete organization_name_list

Usage

The organization delete operation deletes organizations from the registry. The argument is a list of names of organizations being deleted. If a named organization does not exist an error, is generated. An empty string is returned upon successful completion.

This operation also deletes any accounts associated with organizations that are deleted. To preserve accounts, add principals to a different organization using the organization add -member command. Modify the principals' accounts to point to the new organization using the account modify command. Then, you can delete the organization using the organization delete command.

Privilege Required: You must have d (delete) permission to the /./sec/org directory. You must have rD (read, Delete_object) permission on the organization being deleted.

Examples

dcecp> organization delete temps
dcecp>

Related Information

Commands:

account
group
registry
dcecp principal rgyedit
organization help

Returns help information about the organization object and its operations.

Format

organization help [operation | -verbose] [-syntax]

Options

- syntax Displays the syntax diagram of each organization operation
- verbose Displays information about the organization object.

Usage

The organization help command is used without an argument or option to return brief information about each organization operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the organization object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the organization help command.

Examples

dcecp> organization help
add Adds a member to the named organization.
catalog Returns a list of all the names of organizations.
create Creates an organization in the registry.
delete Deletes an organization from the registry.
list Returns a list of all the members of an organization
modify Changes the information about an organization.
remove Removes a member from the named organization.
rename Renames the specified organization.
show Returns the attributes of an organization.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>
organization list

Returns a list of the names of all members of an organization.

Format

organization list organization_name_list [-simplename]

Options

-simplename  organization list  Returns a list of organization names in the registry that are relative, not fully qualified.

Usage

The organization list operation returns a list of the names of all members of an organization. The list operation uses no arguments. By default, fully qualified names are returned in the form cellname/organizationname. If the -simplename option is given, the cell name is relative, not fully qualified.

Privilege Required:  You must have r (read) permission to the organization.

Examples

dcecp> organization list managers
/.../mycell.goodco.com/W_Ward
/.../mycell.goodco.com/L_Jones
/.../mycell.goodco.com/S_Preska
/.../mycell.goodco.com/S_Rohrer

dcecp>

dcecp> organization list {managers users}
/.../mycell.goodco.com/W_Ward
/.../mycell.goodco.com/L_Jones
/.../mycell.goodco.com/S_Preska
/.../mycell.goodco.com/S_Rohrer
/.../mycell.goodco.com/W_Rosenberry
/.../mycell.goodco.com/J_Severance
/.../mycell.goodco.com/J_Hunter
/.../mycell.goodco.com/B_Carr
/.../mycell.goodco.com/E_Vliet

dcecp>

Related Information

Commands:

account  group  registry

dcecp  principal  rgyedit
organization modify

Changes attributes and policies of organizations.

Format

```
organization modify organization_name_list
[[-change attribute_list | attribute options] |
 -add extended_registry_attribute_list |
 -remove extended_registry_attribute_list [-types]]
```

Options

- **-add extended_registry_attribute_list**
  Lets you add extended registry attributes that may be defined for your environment. You can specify the attributes being added as a list of one or more extended registry attributes. See the [z/OS DCE Administration Guide](#) for more information about extended registry attributes.

- **-change attribute_list**
  Lets you specify attributes using an attribute list (see “Attributes” on page 279). For example, the format is:

  `[-change {fullname string} {acctlife relative_time} {pwdalpha string}]`

  The `-change` option is intended for use in scripts when you can paste in lengthy attribute lists output by previous commands. The alternative attribute options may be easier to use for interactive commands. Note that you can modify ERAs by including them in an `attribute_list`. You cannot specify ERAs as individual attribute options.

- **-remove extended_registry_attribute_list**
  Lets you remove extended registry attributes that may be defined for your environment. You can specify the attributes being removed as a list of one or more extended registry attributes. See the [z/OS DCE Administration Guide](#) for more information about extended registry attributes.

- **-types**
  Used with the `-remove` option to specify that the value of the `-remove` option is a list of attribute types. This means that the entire attribute is removed. This option is not valid without the `-remove` option.

Alternative attribute options you can modify are:

- **-acctlife {relative_time | unlimited}**
  This policy attribute defines the life span of accounts. Specify the time using the DTS relative time format (`[-]DD-hh:mm:ss`) or the string `unlimited`.

- **-fullname string**
  This attribute is used for information purposes only. It usually describes or expands a primary name to allow easy recognition by users. For example, an organization can have a primary name of `osf` and a full name of `Open Software Foundation`. The value is a string. If it contains spaces, it displays in quotation marks, and must be in quotation marks or braces (as per Tcl quoting rules) when entered. If not entered, the full name defaults to the null string (that is, blank).

- **-pwdalpha {yes | no}**
  This policy attribute defines whether passwords can consist entirely of alphanumerical characters. Its value is either `yes` or `no`.


-pwdexpdate date
This policy attribute defines a date when a password expires. Specify the time using an
ISO compliant time format (such as (CC-MM-DD-hh:mm:ss) or the string none in which
case, there is no expiration date for the password.

-pwdlife {relative_time | unlimited}
This policy attribute defines the lifespan of passwords. Specify the time using the DTS
relative time format ([DD-hh:mm:ss]) or the string unlimited.

-pwdminlen integer
This policy attribute defines the minimum number of characters in a password. Its value
is a positive integer or the integer 0, meaning there is no minimum length checking.

-pwdspaces {yes | no}
This policy attribute defines whether passwords can consist entirely of spaces. Its value
is either yes or no.

Usage
The organization modify operation changes attributes and policies of organizations. (To change
registry-wide policies, use the registry command.)

The argument is a list of names of organizations being operated on. All modifications are applied to all
organizations named in the argument. Organizations are modified in the order they are listed and all
modifications to an individual organization are atomic. Modifications to multiple organizations are not
atomic. A failure for any one organization in a list generates an error and the rest of the operation is
halted. An empty string is returned upon successful completion.

The -change option can be used to modify the value of any one of the attributes except for orgid and
uuid or any extended registry attribute. The value of the -change option is an attribute list describing the
new values for the specified attributes. Or, you can use attribute options of the form -acctlife
{relative_time | unlimited}.

A maximum of 256 values are allowed in the extended_registry_attribute_list. Specify only ERA attributes
when using the -add and -remove options.

Privilege Required: You must have rm (read, mgmt_info) permission to the organization being
modified.

Examples

dcecp> organization modify temps -acctlife 180-00:00:00 \  
> -pwdlif 30-00:00:00 -pwdexpdate 1997-12-31-23:59:59

dcecp>

Related Information

Commands:

account group registry
dcecp principal rgyedit
organization operations

Returns a list of the operations supported by the organization object.

**Format**

organization operations

**Usage**

The organization operations command uses no arguments, and returns a list of the available operations for the organization object. The order of the elements is alphabetic, except that help and operations are listed last.

**Privilege Required:** No special privileges are needed to use the organization operations command.

**Examples**

dcecp> organization operations
add catalog create delete list modify remove rename show help operations
dcecp>

**Related Information**

Commands:

- account
- dcecp
- group
- principal
- registry
- rgyedit
organization remove

Removes a member from an organization.

Format

organization remove organization_name_list -member member_name_list

Options

- -member member_name_list
   A list of one or more names of principals being removed from each organization in the argument.

Usage

The organization remove operation removes a member from an organization. The argument is a list of names of organizations from which to remove members. The value of the required -member option is a list of names of principals to be removed from the organizations listed in the argument. An empty string is returned upon successful completion.

This operation also deletes accounts associated with the principals being removed. To preserve any accounts, first add principals to a different organization using the organization add -member command. Modify the principals’ accounts to point to the new organization using the account modify command. Then you can remove the members from the organization using the organization remove command.

Privilege Required: You must have rM (read, Member_list) permission on the target organizations and r (read) permission on the member being removed.
organization remove

Examples

dcecp> organization remove managers -member J_Wanders

dcecp> organization add rigel -member W_Rosenberry

dcecp> account modify W_Rosenberry -organization rigel

dcecp> account show W_Rosenberry
{acctvalid yes}
{client yes}
{created /.../mycell.goodco.com/cell_admin 1996-06-30-12:39:48.000+00:00I-----}
{description {}}
{dupkey no}
{expdate none}
{forwardabletkt yes}
{goodsince 1994-06-30-12:39:48.000+00:00I-----}
{group users}
{home /}
{lastchange/.../mycell.goodco.com/cell_admin 1996-06-30-12:39:48.000+00:00I-----}
{organization rigel}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}
dcecp> organization remove gemini -member W_Rosenberry

dcecp>

Related Information

Commands:

account
dcecp
group
principal
registry
rgyedit
organization rename

Changes the name of a specified organization.

Format

organization rename organization_name -to new_organization_name

Options

-to new_organization_name

The -to option specifies the new name of the organization.

Usage

The organization rename operation changes the name of a specified organization. The argument is a single name of an organization being renamed. This command uses a required -to option with a value of the new name. The value may not be a list. An empty string is returned upon successful completion.

Privilege Required: You must have rf (read, fullname) permission to the specified organizations.

Examples

dcecp> organization list rigel
../../mycell.goodco.com/H_Lichtin
../../mycell.goodco.com/R_Mathews
../../mycell.goodco.com/S_Teto
../../mycell.goodco.com/J_Shirley
../../mycell.goodco.com/K_Digan
../../mycell.goodco.com/W_Rosenberry
../../mycell.goodco.com/W_Williams
../../mycell.goodco.com/D_Weir
../../mycell.goodco.com/L_Zahn
../../mycell.goodco.com/P_Neilson
dcecp> organization rename rigel -to sirus
dcecp> organization list rigel
EUVARzerodot7172E DCE interface error.

DCE status code: 0x1712207a - Requested object does not exist in user registry database.
dcecp> organization list sirus
../../mycell.goodco.com/H_Lichtin
../../mycell.goodco.com/R_Mathews
../../mycell.goodco.com/S_Teto
../../mycell.goodco.com/J_Shirley
../../mycell.goodco.com/K_Digan
../../mycell.goodco.com/W_Rosenberry
../../mycell.goodco.com/W_Williams
../../mycell.goodco.com/D_Weir
../../mycell.goodco.com/L_Zahn
../../mycell.goodco.com/P_Neilson
dcecp>
organization rename

Related Information

Commands:

- account
- dcecp
- group
- principal
- registry
- rgyedit
organization show

Shows registry information for the specified organizations.

Format

```
organization show organization_name_list [-policies] [-xattrs] [-all]
```

Options

- `-all` Returns the attributes followed by the ERAs and policies.
- `-policies` Returns only the policies of the organization, with no other attributes.
- `-xattrs` Returns only the ERAs (Extended Registry Attributes) of the organization, with no other attributes.

Format

The `organization show` operation returns an attribute list describing the specified organizations. The argument is a list of names of organizations being operated on. If more than one organization is given, the attributes are concatenated.

Attributes are returned in the following order:

- `fullname`
- `orgid`
- `uuid`

Policies are returned in the following order:

- `acctlife`
- `pwdalpha`
- `pwdexpdate`
- `pwdlife`
- `pwdminlen`
- `pwdspaces`

If the organization does not have any policies, then `No policy` is returned.

The policies that are actually in effect can be different from the organization policies if the registry-wide policies are more restrictive. If this is the case, the show command alters the attribute structure on output to include an `effective` tag and the effective value. For example:

```
dcecp> org show foo -policies
{acctlife 30 days}
{pwdalpha no}
{pwdexpdate none}
{pwdlife unlimited effective 5 days}
{pwdminlen 6}
{pwdspaces no}
dcecp>
```

**Privilege Required:** You must have `r` (read) permission to the specified organizations.
organization show

Examples

dcecp> organization show temps
{fullname {Temporary Employees}}
{orgid 103}
{uuid 00000067-9402-21cd-a602-00000c08adf6}
dcecp>

dcecp> organization show temps -policies
{acctlife +180-00:00:00.000I----}
{pwdalpha yes}
{pwdexpdate 1997-12-31-23:59:59.000+00:00I----}
{pwdlife +30-00:00:00.000I----}
{pwdminlen 0}
{pwdspaces yes}
dcecp>

dcecp> organization show temps -all
{fullname {Temporary Employees}}
{orgid 103}
{uuid 00000067-9402-21cd-a602-00000c08adf6}

{acctlife +180-00:00:00.000I----}
{pwdalpha yes}
{pwdexpdate 1997-12-31-23:59:59.000+00:00I----}
{pwdlife +30-00:00:00.000I----}
{pwdminlen 0}
{pwdspaces yes}
dcecp>

Related Information

Commands:

account
dcecp
group
principal
registry
rgyedit
A dcecp object that manages a principal in the DCE Security Service.

Format

```
principal catalog [server_name_list] [-simplename]
principal create principal_name_list [-attribute attribute_list | attribute options]
principal delete principal_name_list
principal help [operation | -verbose] [-syntax]
principal modify principal_name_list
   {[-change attribute_list | attribute options] | 
   -add extendedregistryattribute_list | 
   -remove extendedregistryattribute_list [-types]}
principal operations
principal rename principal_name -to new_principal_name
principal show principal_name_list [-xattrs] [-all]
```

Parameters

- **operation**: The name of one specific `principal` operation (subcommand) about which you want help information.
- **principal_name**: The name of one principal being renamed.
- **principal_name_list**: A list of one or more names of principals to act on. The names can all be fully qualified principal names as in:
  ```
  /.../cellname/principal_name
  ```
  Or, they can all be cell-relative principal names as in:
  ```
  principal_name
  ```
  Cell-relative names refer to a principal in the cell named in the _s(sec) convenience variable or in the default cell of the local host if the _s(sec) convenience variable is not set.
  Do not mix fully-qualified names and cell-relative names in a list. And, do not use names of registry database objects that contain registry information about principals (principal names cannot begin with ./sec/principal/).

- **server_name_list**: A list of one or more names of registry servers to act on. The names can be fully qualified as in:
  ```
  /.../cellname/subsys/dce/sec/registry_name
  ```
  Or, they can be cell-relative names as in:
  ```
  subsys/dce/sec/registry_name
  ```

Chapter 2. DCE Administration Commands
principal

Usage

The principal object represents registry principals. Unless otherwise noted, all of the operations of this object take one argument which is a list of one or more names of principals to act on. They must be principal names, not the names of the registry database objects that contain registry information about principals (that is, the names cannot begin with /:/sec/principal/).

After this command runs, the _b(sec) convenience variable is set to the name of the server that was bound to for the command. The value of the variable _s(sec) before the command is treated as a hint; the server specified is contacted if it can service the request. A case where it cannot service the request is if a read-only registry was bound to, and the next command is a create command. In this case, the master registry will be bound to automatically and the _b(sec) variable updated appropriately. The value of the variable is the name of the registry bound to in one of the formats specified as valid for the argument to the registry object.

Attributes

alias value

Used with the create and modify operations, the value of this attribute is either yes or no. Each principal can have only one name, but may have one or more alias names. All these names refer to the same principal, and, therefore, the same UUID and uid. While aliases refer to the same principal, they are separate entries in the registry database. Therefore, the instance name given to a principal command can refer to either the primary name or an alias name of a principal. The value of this attribute determines this.

fullname string

Used with the create and modify operations, this specifies the full name of the principal. It is for information purposes only. It usually describes or expands a primary name to allow easy recognition by users. For example, a principal can have a primary name of jsbach and a full name of Johann S. Bach. The value is a string. If it contains spaces, it displays in quotation marks, and must be in quotation marks or braces (as per Tcl quoting rules) when entered. If not entered, the full name defaults to the null string (that is, blank).

quota quota

Used with the create and modify operations to specify the principal object creation quota. This is the total number of registry objects that can be created by the principal. It is either a non-negative number or the string unlimited. A value of 0 prohibits the principal from creating any registry objects. Each time a principal creates a registry object, this value is decremented for that principal.

uuid hexadecimal number

Used with the create operation, this specifies an internal identifier for a principal. Two principals cannot have the same UUID.

uid value

Used with the create operation, this specifies the User Identifier for the principal. Two principals cannot have the same uid, however, aliases can share one uid. It is an integer, and it is often called the UNIX ID.

See the z/OS DCE Administration Guide for more information about attributes.

Related Information

Commands:

account  group  registry
dcecp  organization  rgyedit
principal catalog

Returns a list of the names of all principals in the registry.

Format

```
principal catalog [server_name_list] [-simplename]
```

Options

- `-simplename` Returns a list of principal names in the registry that are relative, not fully qualified.

Usage

The `principal catalog` operation returns a list of the names of all principals in the registry. This command uses an optional server name argument to specify the server to use. By default, fully qualified names are returned in the form `cellname/principalname`. If the `-simplename` option is given, then the principal name is relative, not fully qualified.

Privilege Required: You must have `r (read)` permission to the `/./sec/principal` directory.

Examples

```
dcecp> principal catalog
/.../small_cell.goodco.com/nobody
/.../small_cell.goodco.com/root
/.../small_cell.goodco.com/daemon
/.../small_cell.goodco.com/sys
/.../small_cell.goodco.com/bin
/.../small_cell.goodco.com/uucp
/.../small_cell.goodco.com/who
/.../small_cell.goodco.com/mail
/.../small_cell.goodco.com/tcb
/.../small_cell.goodco.com/dce-ptgt
/.../small_cell.goodco.com/dce-rgy
/.../small_cell.goodco.com/cell_admin
/.../small_cell.goodco.com/krbtgt/small_cell.goodco.com
/.../small_cell.goodco.com/hosts/pmin17/self
/.../small_cell.goodco.com/hosts/pmin17/cds-server
/.../small_cell.goodco.com/hosts/pmin17/gda
/.../small_cell.goodco.com/William_Ward
/.../small_cell.goodco.com/John_Hunter
```

Related Information

Commands:

- `account`
- `dcecp`
- `group`
- `organization`
- `registry`
- `rgyedit`
principal create

Creates a new principal in the registry database.

Format

principal create principal_name_list [-attribute attribute_list | attribute options]

Options

-attribute attribute_list

Lets you specify attributes using an attribute list rather than using the individual attribute options such as alias, uid, and fullname. The format is:

-attribute {{alias value} {uid value} {fullname value}}

The -attribute option is intended for use in scripts when you are pasting in lengthy attribute lists output by previous commands. The individual attribute options may be easier to use for interactive commands. Note that you can specify extended registry attributes (ERAs) with the -attribute option. You cannot specify ERAs as individual attribute options.

The alternative attribute options are:

-alias value

Specifies whether the principal name is an alias. The value of this attribute is either yes or no. Each principal can have only one name, but may have one or more alias names. All of these names refer to the same principal and, therefore, they share the same UUID and uid. While aliases refer to the same principal, they are separate entries in the registry database. Therefore, the instance name given to a principal command can refer to either the primary name or an alias name of a principal. This is determined by the value of the alias attribute.

-fullname string

Specifies the full name of the principal. It is for information purposes only. It usually describes or expands a primary name to be easily recognized by users. For example, a principal can have a primary name of jsbach and a full name of Johann S. Bach. The value is a string. If it contains spaces, it displays in quotation marks, and must be in quotation marks or braces (following dcecp quote rules) when entered. If not entered, the full name defaults to the null string (that is, blank).

-quota quota

Specifies the principal's object creation quota. This is the total number of registry objects that can be created by the principal. It is either a non-negative number or the string unlimited. A value of 0 prohibits the principal from creating any registry objects. Each time a principal creates a registry object, this value is decremented for that principal.

-uuid hexadecimal number

Specifies an internal identifier for a principal. Two principals cannot have the same UUID.

This option is only used to adopt an orphaned UUID. The value of this option is a UUID. Usually, the UUID for a new principal is generated by the registry. In cases where data exists tagged with a UUID of a principal that was deleted from the registry, this option can be used on the create command to specify the old UUID for a new principal. The UUID specified must be an orphan (a UUID for which no name exists in the registry). An error occurs if you specify a name or uid that is already defined in the registry.
The -alias option may not be used with this option, but both the -fullname and -quota options may be.

If the principal is a cell principal, the -uid option can be used to specify the UNIX number to associate with the principal. (If you do not enter this option for a cell principal, the next sequential UNIX number is supplied as a default by the registry.) For all principals other than cell principals, the uid is extracted from information embedded in the UUID of the principal and cannot be specified here.

-uid value  Specifies the User Identifier for the principal. It is an integer and often called the Unix ID. Two principals cannot have the same uid, so do not use this option when creating more than one principal with a single create command. However, aliases can share the same uid.

Usage

The principal create operation creates a new principal in the registry database. The argument is a list of names of principals being created. An empty string is returned upon successful completion. Options specify the attributes of the newly created principal. All options are applied to all principals in the argument.

Note:  The principal name to be created cannot contain the characters @ (at sign) or : (colon).

Privilege Required:  You must have i (insert) permission to the /.:sec/principal directory.

Examples

The following command creates an alias postmaster for the principal with uid 1234:

dcecp> principal create postmaster -uid 1234 -alias yes
dcecp>

Related Information

Commands:

account  group  registry
dcecp  organization  rgyedit
principal delete

Deletes principals from the registry.

Format

`principal delete principal_name_list`

Usage

The `principal delete` operation deletes principals from the registry. When a principal is deleted, the principal account is deleted as well. The argument is a list of names of principals being deleted. Note that these names can be either primary or alias names. In either case, an account associated with that name is deleted. If a named principal does not exist, an error is generated. An empty string is returned upon successful completion.

**Privilege Required:** You must have `d (delete)` permission to the `./sec/principal` directory. You must have `rD (read, Delete_object)` permission on the principal being deleted.

Examples

```
dcecp> principal delete ./William_Rosenberry
```

Related Information

Commands:

- `account`
- `group`
- `organization`
- `registry`
- `rgyedit`
principal help

Returns help information about the principal object and its operations.

Format

principal help [ operation | -verbose ] [-syntax]

Options

-syntax Displays the syntax diagram of each principal operation.
-verbose Displays information about the principal object.

Usage

The principal help command is used without an argument or option to return brief information about each principal operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the principal object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required:  No special privileges are needed to use the principal help command.

Examples

dcecp> principal help
catalog  Returns all the names of principals in the registry.
create   Creates a DCE principal.
delete   Deletes a principal from the registry.
modify   Changes the information about a principal.
rename   Renames the specified principal.
show     Returns the attributes of a principal.
help     Prints a summary of command-line options.
operations Returns the valid operations for command.

Related Information

Commands:

account   group       registry
 dcecp    organization rgyedit
**principal modify**

Changes attributes of principals.

**Format**

```
principal modify principal_name_list
([-change attribute_list | attribute options] |
-add extended_registry_attribute_list |
-remove extended_registry_attribute_list [-types])
```

**Options**

- **-add extended_registry_attribute_list**
  Lets you add extended registry attributes defined for your environment. You can specify the attributes being added as a list of one or more extended registry attributes. See the [z/OS DCE Administration Guide](#) for more information about extended registry attributes.

- **-change attribute_list**
  Lets you modify attributes using an attribute list (see “Attributes” on page 298). The -change option is intended for use in scripts when pasting in lengthy attribute lists output by previous commands. The alternative attribute options might be easier to use for interactive commands. Note that you can modify ERAs by including them in an attribute_list. You cannot specify ERAs as individual attribute options.

- **-remove extended_registry_attribute_list**
  Lets you remove extended registry attributes defined for your environment. You can specify the attributes being removed as a list of one or more extended registry attributes. See the [z/OS DCE Administration Guide](#) for more information about extended registry attributes.

- **-types**
  Used with the -remove option to specify that the value of the -remove option is a list of attribute types. This means that the entire attribute is removed. This option is not valid without the -remove option.

The alternative attribute options are:

- **-alias value**
  Specifies whether the principal name is an alias. The value of this attribute is either yes or no. Each principal can have only one name, but may have one or more alias names. All these names refer to the same principal and, therefore, the same UUID and uid. While aliases refer to the same principal, they are separate entries in the registry database. Therefore, the instance name given to a principal command can refer to either the primary name or an alias name of a principal. This is determined by the value of this attribute.

- **-fullname string**
  Specifies the full name of the principal. It is for information purposes only. It usually describes or expands a primary name to allow easy recognition by users. For example, a principal can have a primary name of jsbach and a full name of Johann S. Bach. The value is a string. If it contains spaces, it displays in quotation marks, and must be in quotation marks or braces (as per Tcl quoting rules) when entered. If not entered, the full name defaults to the null string (that is, blank).

- **-quota quota**
  Specifies the principal’s object creation quota. This is the total number of registry objects that can be created by the principal. It is either a non-negative number or the string unlimited. A value of 0 prohibits the principal from creating any registry objects.
Each time a principal creates a registry object, this value is decremented for that principal.

**Usage**

The `principal modify` operation changes attributes of principals. The argument is a list of names of principals being operated on. All modifications are applied to all principals named in the argument. Principals are modified in the order they are listed and all modifications to an individual principal are atomic. Modifications to multiple principals are not atomic. A failure for any one principal in a list generates an error, and the rest of the operation is halted. An empty string is returned upon successful completion.

The `-change` option modifies the value of any one of the attributes except for `uid` and `uuid`. The value of the `-change` option is an attribute list describing the new values for the specified attributes. It also supports the following attribute options:

- `-alias`
- `-quota`
- `-fullname`

A maximum of 256 values are allowed in the `extended_registry_attribute_list`. Specify only ERA attributes when using the `-add` and `-remove` options.

**Privilege Required:** You must have `rfmu (ready, fullname, mgmt_info, user_info)` permission to the principal being modified.

**Examples**

```bash
dcecp> principal modify /.:joe -fullname "Joe Long"
dcecp> principal show /.:joe
{fullname {Joe Long}}
{uid 30014}
{uuid 0000753e-f51f-2e0e-b000-0000c08adf56}
{alias no}
{quota unlimited}
{groups staff}
dcecp>
```

The following example deletes the attribute names `MVSname` and `MVSinteger` from the principal `joe`:

```bash
dcecp> principal modify /.:joe -remove {MVSname MVSinteger} -types
```

**Related Information**

Commands:

- `account`
- `group`
- `registry`
- `dcecp`
- `organization`
- `rgyedit`
principal operations

Returns a list of the operations supported by the principal object.

Format

principal operations

Usage

The principal operations command uses no arguments, and returns a list of the available operations for the principal object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the principal operations command.

Examples

dcecp> principal operations
catalog create delete modify rename show help operations
dcecp>

Related Information

Commands:

account  group  registry

dcecp  organization  rgyedit
principal rename

Changes the name of a specified principal.

Format

`principal rename principal_name -to new_principal_name`

Options

- `-to new_principal_name`
  Specifies the new name of the principal.

Usage

The `principal rename` operation changes the name of a specified principal. The argument is a single name of a principal being renamed. This command uses a required `-to` option with a value of the new name. The value cannot be a list. An empty string is returned upon successful completion.

**Note:** The principal name to be created cannot contain the characters `@` (at sign) or `:` (colon).

**Privilege Required:** You must have `rf` (read, fullname) permission to the registry object for the specified principal.

Examples

```
dcecp> acl show ./sec/principal/bob
{unauthenticated r--------g}
{user_obj r--------ug}
{user cell_admin rcDnfmaug}
{group acct-admin rcDnfmaug}
{other_obj r--------g}
{any_other r--------}
dcecp> prin rename bob -to joe
EUVA07172E DCE interface error.
    DCE status code: 0x17122081 - User not authorized to access the record.
dcecp> login cell_admin
Enter Password:
dcecp> principal rename bob -to joe
```

Related Information

Commands:

- `account`
- `group`
- `registry`
- `dcecp`
- `organization`
- `rgyedit`
principal show

Shows registry information for the specified principals.

Format

```
principal show principal_name_list [-xattrs] [-all]
```

Options

- **-all**
  Returns the attributes followed by the ERAs.
- **-xattrs**
  Returns only the ERAs (Extended Registry Attributes) of the principal, with no other attributes.

Usage

The `principal show` operation returns an attribute list describing the specified principals. The argument is a list of names of principals being operated on. If more than one principal is given, the attributes are concatenated with a blank line between principals. There is one attribute in addition to `fullname`, `uid`, `uuid`, `alias` and `quota`. It is called `groups`, and its value is a list of group names that the principal is a member of. Attributes are returned in the following order:

- `fullname`
- `uid`
- `uuid`
- `alias`
- `quota`

followed by `groups`.

If called with the `-xattrs` option, the Extended Registry Attributes (ERAs) are returned instead of the above attributes. If called with `-all`, both are returned.

**Privilege Required:** You must have `r` (read) permission to the specified principals.

Examples

```
dcecp> principal show ./:/joe
{fullname {Joe Long}}
{uid 30014}
{uuid 0000753e-f51f-2e0-e-b000-0000c08ad5f6}
{alias no}
{quota unlimited}
{groups staff}
dcecp>
```

Related Information

Commands:

```
account  group  registry
dcecp  organization  rgyedit
```
### registry

A `dcecp` object that manages a registry in the DCE Security Service.

### Format

```
registry catalog [cell_name]
registry connect cell_name
  -group local_group_name -org local_org_name -mypwd local_password
  -fgroup foreign_group_name -forg foreign_org_name
  -facct foreign_account_name -facctpwd foreign_account_password
  [-expdate account_expiration_date]
registry delete registry_replica_name [-force]
registry designate registry_replica_name [-slave | -master [-force]]
registry destroy registry_replica_name
registry disable [registry_replica_name]
registry dump
registry enable [registry_replica_name]
registry help [operation | -verbose] [-syntax]
registry modify [registry_replica_name] [{-change attribute_list} [attribute options] | -key]
registry operations
registry show [registry_replica_name] [-attributes | -policies | -replica | -master]
registry stop registry_replica_name
registry synchronize registry_replica_name
registry verify
```

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cell_name</code></td>
<td>The name of a cell to act on.</td>
</tr>
<tr>
<td><code>operation</code></td>
<td>The name of one specific registry operation (subcommand) about which you want help information.</td>
</tr>
<tr>
<td><code>registry_replica_name</code></td>
<td>The name of one registry replica to act on. The replica can be a master or a slave replica. The argument, which overrides a value in the _s(sec) convenience variable, can be one of the following:</td>
</tr>
</tbody>
</table>

- A specific cell name to bind to any replica in the named cell; for example, `/.../dcecp.cell.osf.org` or `./` for the local cell.
- The global name of a replica to bind to that specific replica in that specific cell; for example, `./sybsys/dce/sec/master`.
- The name of a replica as it appears on the replica list to bind to that replica in the local cell; for example, `sybsys/dce/sec/master`.
- A string binding to a specific replica; for example, `{ncadg_ip_udp 15.22.144.163}`.

See the Binding Information in the RPC Fundamentals section of the [z/OS DCE Application Development Guide: Core Components](https://www.ibm.com/docs/en/zos/2.4.0?topic=core-components) SC24-5905, for the two
available syntaxes for string bindings. This form is used primarily for debugging or when the Cell Directory Service is not available.

If no argument is given, the value of _s(sec) is used. If the variable is not set, the default argument of /.: is assumed.

Usage

The registry object represents the security service database of account information. The registry is a replicated database: each instance of a registry server, secd, maintains a working copy of the database in virtual memory and on disk. One server, called the master replica, accepts updates and handles the subsequent propagation of changes to all other replicas. All other replicas are slave replicas, which accept only queries. Each cell has one master replica and may have numerous slave replicas. The term replica refers to either a master or slave copy of the registry. The commands in dcecp automatically bind to a registry server capable of performing a specific operation. For most operations, the user does not care which registry server is bound to, as long as the request is fulfilled. However, administrators modifying or checking the replica configuration need to be able to bind to a specific registry. The registry command argument lets you specify which replica to use.

After this command runs, the _b(sec) convenience variable is set to the name of the server that was bound to for the command. All of the commands, except dump and verify, accept an argument and some require one.

Other dcecp objects such as: principal, group, organization, and account represent most of the data in the registry. The registry object is used only for security replica configuration (for example, sec_admin functions) and to manipulate registry-wide attributes (called properties), policies, and registry ERAs. Registry-wide policies refer to policies for organizations (standard policies in rgy_edit terminology) and accounts (see the rgy_edit auth policies command). For a list of the policies supported, see "Attributes."

Two ACLs control access to registry operations. For operations dealing with replication, the replist object ACL (usually /.:/sec/replist) controls access. For those that deal with registry attributes and policies, the policy object ACL (usually /.:/sec/policy) controls access. All of the registry operations described below deal with replication information, and as such, access to them is based on the replist object, except for the modify and show operations when they are dealing with non-replication attributes.

Attributes

The registry supports the following kinds of attributes:

- Modifiable registry attributes that apply to principals, groups, organizations, and accounts created in registries.
- Modifiable registry-wide policy attribute default values. Some of these policy attributes apply to organizations while others apply to accounts. The organization and account objects can override their respective registry-wide policy default values only by setting more restrictive values.
- Synchronization information maintained by each replica. These are read-only attributes that cannot be modified by administrators.
- Replica-specific attributes kept by the master replica for each read-only replica. These are read-only attributes that cannot be modified by administrators.

The registry attributes are:

**deftktlife relative_time**

The default lifetime (in hours) for tickets issued to principals in this cell registry. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss).
registry

**hidepwd {yes | no}**
Determines whether encrypted passwords are displayed or not. If this attribute is set to *yes*, an asterisk (*) displays in place of the encrypted password in command output and in files where passwords display. Possible values are either *yes* or *no*.

**maxuid integer**
The highest number that can be supplied as a *uid* when principals are created. This maximum applies to both the system-generated and user-entered UIDs. The value is an integer.

**mingid integer**
The starting point for GIDs automatically generated by the Security Service when a group is created. You can explicitly enter a lower *gid* than this number; it applies only to automatically generated numbers. The value is an integer.

**minorgid integer**
The starting point for ORGIDs automatically generated by the Security Service when an organization is created. You can explicitly enter a lower *orgid* than this number; it applies only to automatically generated numbers. The value is an integer.

**mintktlife integer**
The minimum amount of time (in minutes) before the principal's ticket must be renewed. The value is an integer. This renewal is performed automatically with no intervention on the part of the user. The shorter this time is, the greater the security of the system. However, extremely frequent renewal can degrade system performance. When selecting the value of this attribute, consider both system performance and the level of security required by the cell. This is a registry-wide value only, it cannot be set for individual accounts.

**minuid integer**
The starting point for *uids* automatically generated by the Security Service when a principal is created. You can explicitly enter a lower *uid* than this number; it applies only to automatically generated numbers. The value is an integer.

The registry-wide policy attributes are:

**acctlife {relative_time | unlimited}**
This registry-wide organization policy defines the life span of accounts. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss) or the string *unlimited*.

**maxtktlife relative_time**
This registry-wide account policy defines the maximum amount of time in hours that a ticket can be valid. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss). When a client requests a ticket to a server, the lifetime granted to the ticket takes into account the *maxtktlife* set for both the server and the client. In other words, the lifetime cannot exceed the shorter of the server's or client's *maxtktlife*.

**maxtktrenew relative_time**
This registry-wide account policy defines the amount of time in hours before a principal ticket-granting ticket expires and that principal must log in again to the system to reauthenticate and obtain another ticket-granting ticket. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss). The lifetime of the principal service tickets can never exceed the lifetime of the principal ticket-granting ticket. The shorter you make this, the greater the security of the system. However, because principals must log in again to renew their ticket-granting ticket, the time needs to consider the user convenience and the level of security required.

**pwdalpha {yes | no}**
This registry-wide organization policy defines whether passwords can consist entirely of alphanumeric characters. Its value is either *yes* or *no*. 
This registry-wide organization policy defines a date on which a password expires. The date is entered as an internationalized date string or the string none, in which case, there is no expiration date for the password.

This registry-wide organization policy defines the life span of passwords. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss) or the string unlimited.

This registry-wide organization policy defines the minimum number of characters in a password. Its value is a positive integer or the integer 0, which means there is no minimum length checking.

This registry-wide organization policy defines whether passwords can consist entirely of spaces. Its value is either yes or no.

The synchronization information maintained by each replica consists of:

A list of the network addresses of the replica. Can be more than one for connectionless and connection-oriented protocols for example.

The name of the cell that the replica is in. It is a fully qualified cell name.

The sequence number of the last update the replica received. A sequence number consists of two 32-bit integers separated by a dot (high.low). The high integer increments when the low integer wraps. An example of this attribute is {lastupdseq Rzerodot.178}.

The localized date and time the replica was last updated.

The network address of the master registry replica of the cell. This is what the replica believes; it is not necessarily correct. Can be more than one for connectionless and connection-oriented protocols for example.

The master sequence number, which is the sequence number of the event that made the replica the master. This is what the replica believes; it is not necessarily correct. A sequence number consists of two 32-bit integers separated by a dot (high.low). The high integer increments when the low integer wraps. An example of this attribute is {masterseqnum Rzerodot.1RzerodotRzerodot}.

The UUID of the master registry replica of the cell. This is what the replica believes; it is not necessarily correct. The value is a UUID.

The name of the replica. It is in the form of a fully qualified CDS name.

The state of the replica, one of:

The replica is in the process of becoming a master.

The replica is a master in the process of becoming a slave.

The replica is in the process of having its master key changed.

The replica is in the process of stopping.

The replica is in the process of initializing (copying its database to) another replica.

The replica is in the process of deleting itself.
disabled The replica is unavailable for updates but will accept queries. This was called in maintenance by sec_admin.
dupmaster Two masters were found in the cell, and the replica is a duplicate of the real master.
enabled The replica is available for use. This was called in service by sec_admin.
initializing The replica is in the process of being initialized by the master replica or another up-to-date replica.
savingdb The replica is in the process of saving its database to disk.
unavailable The replica cannot be reached.
uninitialized The database is a stub database that was not initialized by the master replica or another up-to-date replica.
unknown The replica is not known to the master.
type Specifies if the replica is a master or a slave.
updseqqueue A list of two update sequence numbers that are still in the propagation queue and are not yet propagated. The first number is the base propagation sequence number (the last number known to be received by all replicas). The second number is the sequence number of the last update made on the master. This attribute is only present in the master replica. The sequence numbers consist of two 32-bit integers separated by a dot (high.low). The high integer increments when the low integer wraps. An example of this attribute is {updseqqueue {Rzerodot.1RzerodotRzerodot Rzerodot.178}}.
uuid The UUID of the replica.
version The version of the security server software.

The replica-specific attributes kept by the master replica for each read-only replica are:

lastcommstatus The status of the last communication with the replica. This information is meaningful only if propstatus is update.

lastupdseqsent The last sequence number of the last update sent to this replica. A sequence number consists of two 32-bit integers separated by a dot (high.low). The high integer increments when the low integer wraps. An example of this attribute is {lastupdseqsent 0.175}. This information is meaningful only if propstatus is update.

lastupdttime The localized time the last update was sent to the replica. This information is meaningful only if propstatus is update.

name The name of the replica. It is in the form of a fully qualified CDS name.

numupdtogo The number of outstanding updates. The value is an integer. This information is meaningful only if propstatus is update.

propstatus The status of the propagation, possible values of:
delete The replica is marked for deletion.
initing The replica is in the process of initialization, that is getting an up-to-date copy of the registry.
initmarked The replica is marked for initialization, that is getting an up-to-date copy of the registry.
unavailable The replica cannot be reached.
update The replica is ready to receive propagation updates.
registry

type Specifies whether the replica is a master or a slave.

See the [z/OS DCE Administration Guide] for more information about attributes, policies, and synchronization information.

Related Information

Commands:

```
dcecp principal registry rgyedit
group organization rgyedit secadmin
```
registry catalog

Returns a list of the names of the security servers running in the cell.

Format

registry catalog [cell_name]

Usage

The registry catalog operation returns a list of the names of the security servers (for example, each copy of the registry) running in the cell. This is also known as the replica list. The order of elements returned is arbitrary. The optional argument can specify the name of one other cell or a single string binding. This command sets the _b(sec) variable to the replica to which it binds.

Privilege Required: No special privileges are needed to use the registry catalog command.

Examples

dcecp> registry catalog /.../dcecp.cell.osf.org
/.../dcecp.cell.osf.org/subsys/dce/sec/master
/.../dcecp.cell.osf.org/subsys/dce/sec/ice
dcecp>

Related Information

Commands:

dcecp  principal  registry  secadmin
group  organization  rgyedit  xattrschema
registry connect

Connects the local (that is, default) cell of the local host to the foreign cell specified by the argument.

Format

```
registry connect cell_name
    -group local_group_name -org local_org_name -mypwd local_password
    -fgroup foreign_group_name -forg foreign_org_name
    -facct foreign_account_name -facctpwd foreign_account_password
    [-expdate account_expiration_date]
```

Options

- `-expdate account_expiration_date`
  Optionally sets an expiration date for both local and foreign accounts.

- `-facct foreign_account_name`
  Specifies the name for the administrator in the foreign cell.

- `-facctpwd foreign_account_password`
  Specifies the password for the administrator in the foreign cell.

- `-fgroup foreign_group_name`
  Specifies the group for the foreign account that is being created.

- `-forg foreign_org_name`
  Specifies the organization for the foreign account that is being created.

- `-group local_group_name`
  Specifies the group for the local account that is being created.

- `-mypwd local_password`
  Specifies the password for the administrator in the local cell.

- `-org local_org_name`
  Specifies the organization for the local account that is being created.

Usage

The `registry connect` operation creates an account in the local cell for the specified foreign cell and also creates an account in the foreign cell for the local cell. This allows local principals to access objects in the foreign cell as authenticated users, and foreign principals to access objects in the local cell as authenticated users. Both accounts have the same key. The argument must be the fully qualified name of a single cell. It cannot be a list or a string binding.

The principal name created in the local cell is:

```
/.../local_cell_name/sec/principal/krbtgt/foreign_cell_name
```

The principal name created in the foreign cell is:

```
/.../foreign_cell_name/sec/principal/krbtgt/local_cell_name
```

The `-group`, `-org`, and `-mypwd` options supply the account information for the local cell. The `-fgroup`, `-forg`, `-facct`, and `-facctpwd` options supply the account information for the foreign cell.
This command creates the group and organization specified as the values of the relevant options, if necessary, and puts the relevant principal in them, if necessary.

**Note:**

Enter this command and options with care; it creates the group and organization using the exact names you enter.

If the command fails, it removes any organization and group that it created and it also removes the relevant principals.

To protect the password being entered, the `registry connect` command can be entered only from within the `dcecp` program. You cannot enter this command from the operating system prompt using `dcecp` with the `-c` option.

**Privilege Required:** You must have the permissions needed to create the accounts, and A (Admin) permission to the `replist` object.

**Examples**

```
dcecp> registry connect /.../your_cell.com -group none -org none \
> -mypwd -dce- -fgroup none -forg none -facct cell_admin \
> -facctpwd -dce-
```

**Related Information**

Commands:

<table>
<thead>
<tr>
<th>dcecp</th>
<th>principal</th>
<th>registry</th>
<th>secadmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>organization</td>
<td>rgyedit</td>
<td>xattrschema</td>
</tr>
</tbody>
</table>
registry delete

Deletes a registry replica from the cell.

Format

registry delete registry_replica_name [-force]

Options

-force Used when the target replica is not available, the -force option removes the replica name from the replica list of the master replica and propagates the deletion to other replicas that remain on the list.

Usage

The registry delete operation, when called with no options, performs an orderly deletion of a security replica specified as the argument. That is, it binds to the master registry, marks the specified replica as deleted, propagates this deletion to the other replicas, sends a message to the specified replica to delete itself, and finally removes the replica from the master server replica list. The argument is the name of a single security replica.

If the specified replica is unreachable, the -force option can be used. This option causes the master to delete the specified replica from the replica list and then propagates that deletion to the other replicas on the list. The specified replica is not contacted. If the target replica later becomes available, use the registry destroy command to remove the replica from the cell.

Note that the registry delete command only tells the master replica to perform the delete procedure. The command may return before the delete has completed.

An empty string is returned upon successful completion. The command sets _b(sec) to the master replica.

Privilege Required: You must have d (delete) permission to the replist object.

Examples

dcecp> registry delete /./subsys/dce/sec/ice

Related Information

Commands:

dcecp principal registry secadmin
group organization rgyedit xattrschema
registry designate

Changes which replica is the master.

Format

registry designate registry_replica_name [-slave | -master [-force]]

Options

-force Forces the command to set the specified replica as the master, even if other slaves are more up-to-date. Used only with the -master option.

-master Designates the specified replica as the master.

-slave Designates the specified replica as a slave.

Usage

The argument is required and must be the name of a single registry replica. If no options are used, the command sets the named replica as the new master for the cell and sets the existing master as a slave. An empty string is returned upon successful completion.

You can also use the -slave or -master option to explicitly set the named replica as either a slave or the master. However, these options are not recommended because updates can be lost. If you use the -master option, the command checks other slaves to see if they are more up-to-date than the one specified in the argument. If any are, the command fails with an error, unless you specify the -force to override this default action. The command also fails if there is already a master.

The command sets the value of _b(sec) as follows:

- If no options are used, _b(sec) is set to the old master replica.
- If -slave or -master is used, _b(sec) is set to the replica specified in the argument.

Privilege Required: You must have A (Admin) permission to the replist object.

Examples

dcecp> registry designate 
.../dcecp.cell.osf.org/subsys/dce/sec/ice

dcecp>

You can then use the registry dump command to verify that the type attribute changed and that the previous master is now a slave, while the specified replica is now the master.

Related Information

Commands:

dcecp        principal        registry        secadmin
  group       organization     rgyedit         xattrschema
registry destroy

Removes a registry replica from the cell.

Format

registry destroy registry_replica_name

Usage

registry destroy removes the security replica specified as the argument from the cell. The replica destroys its copy of the registry database and stops running. The master registry is not informed of the removal of the replica.

To perform a more orderly deletion of the replica, use the registry delete command. If that command fails because the replica is not available, enter the registry delete command again with the -force option. Then, when the replica becomes available, use registry destroy to remove it from the cell.

An empty string is returned upon successful completion. The command sets _b(sec) to the replica to which it binds.

Privilege Required: You must have d (delete) permission to the replist object.

Examples

dcecp> registry destroy ./:/subsys/dce/sec/ice
dcecp>

Related Information

Commands:

dcecp
  group
  principal
  organization
  registry
  rgyedit
  secadmin
  xattrschema
registry disable

Disables the master registry for updates.

Format

registry disable [registry_replica_name]

Usage

The registry disable operation disables the master registry for updates. Usually this mode is used for maintenance purposes. The argument is a single name of a master registry to be disabled. If no argument is given, the command uses the name in the _s(sec) convenience variable. If no argument is given and the _s(sec) variable is not set, the command uses the master in the local cell.

An empty string is returned upon successful completion, and the command sets _b(sec) to the name of the master replica in the cell.

Privilege Required: You must have A (Admin) permission to the replist object.

Examples

dcecp> registry disable /.../dcecp.cell.osf.org/subsys/dce/sec/master
dcecp>

Related Information

Commands:

dcecp principal registry secadmin
group organization rgyedit xattrschema
registry dump

Returns the replica information for each replica in the cell.

**Format**

registry dump

**Usage**

The *registry dump* command returns the replica information for each replica in the cell. It uses no arguments and, after completing, sets `_b(sec)` to the name of the last replica displayed. Replicas are displayed with a blank line between them. You can set the `_s(sec)` convenience variable to specify another cell or a string binding.

This command is the same as the following script:

```bash
foreach i [registry catalog] {
    lappend r [registry show $i -replica]
    append r
}
return r
```

**Privilege Required:** No special privileges are needed to use the *registry dump* command.
Examples

dcecp> registry dump
{name /.../dcecp.cell.osf.org/subsys/dce/sec/master}
{type master}
{cell /.../dcecp.cell.osf.org}
{uuid a1248a5e-e1e6-11cd-aa0c-0800092734a4}
{status enabled}
{lastupdttime 1996-10-13-14:44:48.000-04:00I-----}
{lastupdseq 0.271}
{addresses
{ncacn_ip_tcp 130.105.5.121}
{ncadg_ip_udp 130.105.5.121}}
{masteraddrs
{ncacn_ip_tcp 130.105.5.121}
{ncadg_ip_udp 130.105.5.121}}
{masterseqnum 0.100}
{masteruuid a1248a5e-e1e6-11cd-aa0c-0800092734a4}
{version secd.dce.1.1}
{updatequeue {0.204 0.271}}

{name /.../dcecp.cell.osf.org/subsys/dce/sec/ice}
{type slave}
{cell /.../dcecp.cell.osf.org}
{uuid c772f46a-e1ec-11cd-9a16-00000c0239a70}
{status enabled}
{lastupdttime 1996-10-13-14:44:48.000-04:00I-----}
{lastupdseq 0.271}
{addresses
{ncacn_ip_tcp 130.105.5.45}
{ncacn_ip_tcp 130.105.5.45}
{ncadg_ip_udp 130.105.5.45}}
{masteraddrs
{ncacn_ip_tcp 130.105.5.121}
{ncadg_ip_udp 130.105.5.121}}
{masterseqnum 0.100}
{masteruuid a1248a5e-e1e6-11cd-aa0c-0800092734a4}
{version secd.dce.1.1}
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>dcecp</th>
<th>principal</th>
<th>registry</th>
<th>secadmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>organization</td>
<td>rgyedit</td>
<td>xattrschema</td>
</tr>
</tbody>
</table>
registry enable

Enables the master registry for updates.

Format

registry enable [registry_replica_name]

Format

The registry enable operation enables the master registry for updates. The argument is a single name of a master registry to be enabled. Read-only copies are not allowed, but the name of other cells may be used. If no argument is given, the command uses the name in the _s(sec) convenience variable. If the _s(sec) variable is not set, the command uses the master in the local cell.

An empty string is returned upon successful completion, and the command sets _b(sec), to the name of the master replica in the cell.

Privilege Required: You must have A (Admin) permission to the replist object.

Examples

dcecp> registry enable .../dcecp.cell.osf.org/subsys/dce/sec/master

dcecp>

Related Information

Commands:

dcecp  principal  registry  secadmin
group  organization  rgyedit  xattrschema
registry help

Returns help information about the registry object and its operations.

Format

registry help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each registry operation.
-verbose Displays information about the registry object.

Usage

The registry help command is used without an argument or option to return brief information about each registry operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the registry object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the registry help command.

Examples

dcecp> registry help
catalog Returns a list of all replicas running in the cell.
connect Creates local and foreign cross-cell authentication accounts.
delete Deletes a replica and removes from master replica list.
designate Changes which replica is the master.
destroy Destroys the specified replica and its copy of the registry database.
disable Disables the specified master registry for updates.
dump Returns replica information for each replica in the cell.
enable Enables the specified master registry for updates.
modify Modifies the master registry or replica.
show Returns attributes of the registry and its replicas.
stop Stops the specified security server process.
synchronize Reinitializes replica with up-to-date copy of the registry
verify Returns a list of replicas not up-to-date with the master.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp principal registry secadmin
group organization rgyedit xattrschema
registry modify

Changes attributes of the registry.

Format

registry modify [registry_replica_name] [-change attribute_list] [attribute options] | -key

Options

- change attribute_list
  Lets you specify attributes using an attribute list (see \textit{Attributes} on page 310). The -change option is intended for use in scripts when you can paste in lengthy attribute lists output by previous commands. The alternative attribute options might be easier to use for interactive commands.

- key
  Generates a new master key for the replica. The -key option requires that \textit{registry_replica_name} be specified.

Usage

The \texttt{registry modify} operation changes attributes of the registry. Modifications always operate on the master registry. The optional argument must name the master registry. If an argument is not supplied, the operation uses the value in the _s(sec) convenience variable as a starting point in searching for the master registry. If no argument is given and the _s(sec) variable is not set, the command uses the master in the local cell. If the -change option is used and the _s(sec) variable is a replica, the command operates on the master of the replica's cell. An empty string is returned upon successful completion, and the command sets the _b(sec) variable to the replica to which it binds.

Use the -change option to modify the value of any one of the attributes using an attribute list format. Or, you can use attribute options of the form -acctlife \{relative_time | unlimited\}

The command also accepts the -key option to generate a new master key for a single replica named in the argument and to re-encrypt that registry's account keys using the new key. The new master key is randomly generated. The -change option and the -key option cannot be used together.

The allowable values on the version attribute for \texttt{registry modify} are \textit{secd.dce.1.1} and \textit{secd.dce.1.2.2}.

Privilege Required: You must have \texttt{rma} (read, management info, authentication info) permission to the policy object.

Examples

\begin{verbatim}
dcecp> registry modify -change {deftktlife +0-08:00:00.000I------}
dcecp>
\end{verbatim}

Related Information

Commands:

\begin{verbatim}
dcecp  principal  registry  secadmin
group   organization  rgyedit  xattrschema
\end{verbatim}
registry operations

Returns a list of the operations supported by the registry object.

Format

registry operations

Usage

The registry operations command uses no arguments, and returns a list of the available operations for the registry object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the registry operations command.

Examples

    dcecp> registry operations
catalog connect delete designate destroy disable dump enable modify show stop
    synchronize verify help operations
dcecp>

Related Information

Commands:

    dcecp  principal  registry  secadmin
    group   organization  rgyedit  xattrschema
registry show

Returns information about the registry and its replicas.

Format

registry show [registry_replica_name] [-attributes | -policies | -replica | -master]

Options

-attributes
Returns an attribute list of the registry-wide attributes followed by any extended registry attributes.

-master
Returns the synchronization information that the master keeps for each slave.

-policies
Returns only the registry-wide polices.

-replica
Returns the synchronization information for the replica.

Usage

The registry show operation returns information about the registry and its replicas. An optional argument specifies a single registry replica to contact. Returns a variety of different information based on the option given. If called with no options or with the -attributes option, returns an attribute list of all the registry-wide attributes followed by any ERAs. If called with the -policies option, returns an attribute list of all the registry-wide polices.

If called with the -replica option, returns the propagation information that is kept by the replica specified in the argument or in _s(sec). The -replica option can be called with an argument that specifies a slave replica. If called in this manner, the master is contacted and the propagation information that the master keeps for the specified slave is returned.

If called with the -master option, returns the propagation information that is kept by the master for each slave.

On success, the command sets _b(sec) to the replica to which it binds.

Privilege Required: No special privileges are needed to use the registry show command.

Examples
dcecp> registry show -attributes
{mingid 31000}
{minorgid 100}
{minuid 30000}
{maxuid 32767}
dcecp> registry show -policies
{deftktlife +0-10:00:00.000I-----}
{mintktlife +0-00:05:00.000I-----}
{hidepwd yes}
dcecp> registry show -replica
{name /.../absolut_cell/subsys/dce/sec/master}
{type master}
{cell /.../absolut_cell}
{uuid 91259b6c-9415-11cd-a7b5-81259b6c-9415-11cd-a7b5-8}
{status enabled}
{lastupdttime 1996-07-05-14:38:15.000-04:00I-----}
{lastupdseq 0.191}
{addresses}
{ncacn_ip_tcp 130.105.5.93}
{ncadg_ip_udp 130.105.5.93}
{masteraddr}
{ncacn_ip_tcp 130.105.5.93}
{ncadg_ip_udp 130.105.5.93}
{masterseqnum 0.100}
{masteruuid 91259b6c-9415-11cd-a7b5-080009251352}
{version secd.dce.1.0.2}
{updsequenque {0.187 0.191}}
dcecp> regi show /.../dcecp.cell.osf.org/subsys/dce/sec/master -master
{name /.../dcecp.cell.osf.org/subsys/dce/sec/master}
{type master}
{name /.../dcecp.cell.osf.org/subsys/dce/sec/ice}
{type slave}
{propstatus update}
{lastupdttime 1996-10-13-14:58:28.000-04:00I-----}
{lastupdseqsent 0.528}
{numupdtogo 0}
{lastcommstatus 0}
dcecp>

Related Information

Commands:

- dcecp
- principal
- organization
- registry
- rgyedit
- secadmin
- xattrschema
registry stop

registry stop

Stops the specified security server process.

Format

registry stop registry_replica_name

Usage

The registry stop operation stops the security server process specified in the argument. The argument is required and must explicitly name one replica. (A cell came is not valid because more than one replica can operate in a cell.) An empty string is returned upon successful completion, and the command sets the _b(sec) convenience variable to the name of the specified replica.

Privilege Required: You must have A (Admin) permission to the replist object.

Examples

dcecp> registry stop ./:/subsys/dce/sec/master

dcecp>

Related Information

Commands:

dcecp     principal
group     organization
registry   rgyedit
secadmin   xattrschema
registry synchronize

Causes the specified replica to reinitialize itself with an up-to-date copy of the database.

Format

registry synchronize registry_replica_name

Usage

The registry synchronize operation causes the specified replica to reinitialize itself with an up-to-date copy of the database. The required argument is a name of a single registry replica to operate on.

This command binds to the master, marks the specified replica for reinitialization, sends a message to the replica informing it to reinitialize itself, and provides a list of other replicas with up-to-date copies of the registry. The specified replica then selects a replica from the list and asks for a copy of the database. Note that the registry synchronize command only starts the synchronize procedure. The command may return before the synchronization has completed.

Usually, you do not need to use the registry synchronize command because registries remain synchronized automatically. If a master loses contact with a slave for a long period of time, the master reinitializes the slave automatically when communication is restored by giving it a fresh copy of the entire database.

An empty string is returned upon successful completion, and the command sets the _b(sec) variable to the name of the master replica contacted.

Privilege Required: You must have A (Admin) permission to the replist object.

Examples

dcecp> registry synchronize /.:/subsys/dce/sec/ice
dcecp>

Related Information

Commands:

  dcecp  principal  registry  secadmin
  group  organization  rgyedit  xattrschema
registry verify

Checks whether all registry replicas are up-to-date.

Format

registry verify

Usage

The `registry verify` command checks whether all registry replicas are up-to-date. If they are, it returns an empty string. If they are not, it returns a list of the out-of-date and unreachable replica names (or bindings if names are not available). You can use the elements of this list in a `foreach` loop around a call to `registry synchronize` to reinitialize each replica. You can set the `_s(sec)` convenience variable to specify another cell or a string binding.

Privilege Required: No special privileges are needed to use the `registry verify` command.

Examples

If the replica is up-to-date, it returns an empty string, as follows:

dcecp> registry verify
dcecp>

If the replica is not up-to-date, it returns the fully-qualified replica name, as follows:

dcecp> registry verify
/.../cell/subsys/dce/sec/ice
dcecp>

Related Information

Commands:

dcecp group principal registry secadmin
organization rgyedit xattrschema
rpcentry

A dcecp object that manages an RPC name service entry.

Format

rpcentry create entry_name_list
rpcentry delete entry_name_list
rpcentry export entry_name_list
\{[-interface interface_id -binding protocol_sequence_list] [-object object_uuid_list]\}
rpcentry help [operation | -verbose] [-syntax]
rpcentry import entry_name_list -interface interface_id [-object object_uuid] [-max integer] [-noupdate]
rpcentry operations
rpcentry show entry_name_list
[-interface interface_id_list] [-object object_uuid_list] [-noupdate]
rpcentry unexport entry_name_list
\{[-interface interface_id [-version versions]] [-object object_uuid_list]\}

Parameters

entry_name_list Specifies a list of one or more names of the target name service entry. For an entry in the local cell, you can omit the cell name and specify only cell-relative names.

operation The name of one specific rpcentry operation (subcommand) about which you want help information.

Usage

The rpcentry object represents an RPC server entry in the cell name service. Use the rpcentry commands to create, modify, display, and delete name service entries.

Related Information

Commands:
dcecp rpccp rpcentry rpccprofile
rpcentry create

Creates an empty entry in the name service.

Format

rpcentry create entry_name_list

Usage

The rpcentry create operation creates an empty entry in the name service. Because an empty entry is the same as an empty RPC group or RPC profile, calling rpcentry create is the same as calling rpcgroup create or rpcprofile create. The entry_name_list argument is a list of names of RPC entries being created. An empty string is returned upon successful completion. If the RPC entry already exists, create returns an error message.

Privilege Required: To create an rpcentry, you need i (insert) permission to the parent directory and both r (read) permission and w (write) permission to the CDS object entry (the target name service entry).

Examples

The following command adds a non-specialized entry to the name service database:

dcecp> rpcentry create ./:LandS/anthro/Cal_host_2

dcecp>

Related Information

Commands:

dcecp endpoint rpccp rpcentry rpcprofile

rpcentry delete

Removes the specified entry from the name service.

Format

**rpcentry delete** `entry_name_list`

Usage

The **rpcentry delete** operation removes the specified entry from the name service. This removes all **group** and **profile** members from the entry and deletes the entry itself. The argument is a list of one or more names of entries being deleted. An empty string is returned upon successful completion. If the entry does not exist, an error is returned.

**Privilege Required:** To delete an entry, you need **r (read)** permission to the CDS object entry (the target name service entry). You also need **d (delete)** permission to the CDS object entry or to the parent directory.

Examples

The following command removes the entry `./LandS/anthro/Cal_host_2` from the local cell of the name service database:

```
dcecp> rpcentry delete ./LandS/anthro/Cal_host_2
```

Related Information

Commands:

```
dcecp       rpccp       rpcentry
endpoint                  rpcprofile
```
rpcentry export

Transfers information to the specified entry in the name service.

Format

rpcentry export entry_name_list
[[-interface interface_id -binding protocol_sequence_list] [-object object_uuid_list]]

Options

-binding protocol_sequence_list
An RPC binding that describes a server location. Declares a list of one or more protocol sequences (RPC bindings). To use this option, you must also specify an interface identifier (using the -interface option). The value has the form of an RPC binding, without an object UUID. The binding information contains an RPC protocol, a network address, and sometimes an endpoint (rpc-prot-seq network-addr endpoint). For a well-known endpoint, include the endpoint in the binding. For example:

-binding {ncadg_ip_udp 63.0.2.17 5347}
For a dynamic endpoint, omit the endpoint from the binding. For example:

-binding {ncadg_ip_udp 63.0.2.17}

-interface interface_id
Declares the interface identifier of one RPC interface. If you specify an interface identifier, you must specify at least one -binding option. The interface identifier has the following form:

(interface-uuid major-version.minor-version)

Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}
Leading zeros in version numbers are ignored.

-object object_uuid_list
A list of object UUIDs. The UUID is a hexadecimal string. Accepts a list of up to 32 object UUIDs. For example:

-object {3c6b8f60-5945-11c9-a236-08002b102989}

Usage

The rpcentry export operation transfers information to the specified entry in the name service. The argument is a list of one or more names of server entries being exported to. If an entry does not exist, it is created. This command uses the -interface, -binding, and -object options to specify what to export. An empty string is returned upon successful completion.

Privilege Required: To export an entry, you need both r (read) permission and w (write) permission to the CDS object entry (the target name service entry). If the entry does not exist, you also need i (insert) permission to the parent directory.
Examples

This example uses the dcecp string syntax to export an rpcentry to CDS:

dcecp> rpcentry export /./subsys/applications/bbs_server \ 
> -interface {458ffcfbe-98c1-11cd-bd93-0000c08adf56 1.0} \ 
> -binding {ncacn_ip_tcp 130.105.1.227} \ 
> -object {76030c42-98d5-11cd-88bc-0000c08adf56}
dcecp>

Related Information

Commands:

- dcecp
- endpoint
- rpccp
- rpcprofile
- rpcentry
rpcentry help

Returns help information about the rpcentry object and its operations.

Format

rpcentry help [operation | -verbose] [-syntax]

Options

- syntax Displays the syntax diagram of each rpcentry operation.
- verbose Displays information about the rpcentry object.

Usage

The rpcentry help command is used without an argument or option to return brief information about each rpcentry operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the rpcentry object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the rpcentry help command.

Examples

dcecp> rpcentry help
create Creates a list of empty RPC Entries.
delete Deletes a list of RPC Entries.
export Stores bindings in a list of RPC Entries.
import Returns the bindings from a list of RPC Entries.
show Returns the attributes of a list of RPC Entries.
unexport Deletes bindings from a list of RPC Entries.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp rpccp rpcentry rpcprofile
rpcentry import

Returns a string binding from the specified RPC entry.

Format

```
rpcentry import entry_name_list -interface interface_id [-object object_uuid]
[-max integer] [-noupdate]
```

Options

- **-interface interface_id**
  Declares the interface identifier of one RPC interface. The interface identifier has the following form:

  \{interface-uuid major-version.minor-version\}

  Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

  `-interface \{ec1ebe60-5943-11c9-a309-08002b102989 3.11\}`

  Leading zeros in version numbers are ignored.

- **-max integer**
  Specifies the maximum number of string bindings to return. A value greater than one returns a list containing up to the number of bindings specified by the value.

- **-noupdate**
  Normally, name service data is cached locally on each machine in a cell. If a name service inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the `-noupdate` option is not specified, `dcecp` goes to a CDS server to retrieve the required data, and updates the local CDS cache. Use the `-noupdate` option to avoid taking the time to update the local cache when you have reason to believe that the local cache is up-to-date.

- **-object object_uuid**
  A list of object UUIDs. The UUID is a hexadecimal string. Accepts a list of up to 32 object UUIDs. For example:

  `-object \{3c6b8f60-5945-11c9-a236-08002b102989\}`

Usage

The `rpcentry import` operation returns a string binding from the specified RPC entry. The argument is a list of names of RPC entries (not a list of RPC entries) to import from. The order of returned bindings is arbitrary.

Privilege Required: You need r (read) permission to the specified CDS object entry (the starting name service entry) and to any CDS object entry in the resulting search path.

Examples

The following command imports a binding:
rpcentry import

dcecp> rpcentry import ./LandS/anthro/Cal_host_3
> -interface {ec1eeb6wzerodot-5943-11c9-a309-08002b02909 1.1}
> -object {30dbeeawzerodot-fb6c-11c9-8eea-08002b0f4528}
{ncacn_ip_tcp 130.105.1.227}
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>dcecp</th>
<th>rppcp</th>
<th>rpcprofile</th>
</tr>
</thead>
<tbody>
<tr>
<td>endpoint</td>
<td>rpcentry</td>
<td></td>
</tr>
</tbody>
</table>
**rpcentry operations**

Returns a list of the operations supported by the `rpcentry` object.

**Format**

```
rpcentry operations
```

**Usage**

The `rpcentry operations` command uses no arguments, and returns a list of the available operations for the `rpcentry` object. The order of the elements is alphabetic, except that `help` and `operations` are listed last.

**Privilege Required:** No special privileges are needed to use the `rpcentry operations` command.

**Examples**

```
dcecp> rpcentry operations
create delete export import show unexport help operations
```

**Related Information**

Commands:

```
dcecp  rpcprof
endpoint  rpcp
           rpcentry
```
rpcentry show

Returns a list containing the binding information in the specified RPC entries.

Format

```
rpcentry show entry_name_list [-interface interface_id_list] [-object object_uuid_list] [-noupdate]
```

Options

- **interface interface_id**
  Declares a list of one or more interface identifiers of RPC interfaces. The interface identifier has the following form:

  `{interface-uuid major-version.minor-version}`

  Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

  `-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}`

  Leading zeros in version numbers are ignored.

- **noupdate**
  Normally, name service data is cached locally on each machine in a cell. If a name service inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the -noupdate option is not specified, dcecp goes to a CDS server to retrieve the required data, and updates the local CDS cache. Use the -noupdate option to avoid taking the time to update the local cache when the local cache is up-to-date.

- **object object_uuid**
  A list of object UUIDs. Accepts a list of up to 32 object UUIDs. The UUID is a hexadecimal string. For example:

  `-object {3c6b8f60-5945-11c9-a236-08002b102989}`

Usage

The rpcentry show operation returns a list containing the binding information in the specified RPC entry. The argument is a list of one or more names of server entries to return information about.

The returned list consists of two lists. Each item in the first list is also a list, where the first two elements are the interface identifier (the UUID and then the version), and the rest of the elements are string bindings in Tcl syntax. The second list is a list of Object UUIDs exported by the server. The order of the data returned is arbitrary.

Privilege Required: You need r (read) permission to the CDS object entry (the target name service entry).

Examples

The following command uses the dcecp string syntax to show the name service entry `/./subsys/applications/bbs_server`.  

```
dcecp> rpcentry show /.:/subsys/applications/bbs_server
{458fffcbe-98c1-11cd-bd93-0000c08adf56 1.0
  {ncacn_ip_tcp 130.105.1.227}}
{76030c42-98d5-11cd-88bc-0000c08adf56}
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>dcecp</th>
<th>rpccp</th>
<th>rpcentry</th>
</tr>
</thead>
<tbody>
<tr>
<td>endpoint</td>
<td></td>
<td>rpcprofile</td>
</tr>
</tbody>
</table>
rpcentry unexport

Removes interface and binding information from an entry in the name service.

Format

rpcentry unexport entry_name_list
[-interface interface_id [-version versions] [-object object_uuid_list]]

Options

-interface interface_id
Declares the interface identifier of an RPC interface. The rpcentry unexport command operates on only one interface_id. If more than one -interface option is specified, the command ignores all but the last interface identifier. The interface identifier has the following form:

{interface-uuid major-version.minor-version}

Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}

Leading zeros in version numbers are ignored.

-object object_uuid
A list of object UUIDs. Accepts a list of up to 32 object UUIDs. The UUID is a hexadecimal string. For example:

-object {3c6b8f60-5945-11c9-a236-08002b102989}

-version versions
Specifies interface version numbers returned with an rpcentry unexport operation using one of the following values for the -version option:

<table>
<thead>
<tr>
<th>Versions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>The interface version is ignored.</td>
</tr>
<tr>
<td>exact</td>
<td>Both the major and minor versions must match the specified versions.</td>
</tr>
<tr>
<td>compatible</td>
<td>The major version must match the specified version, and the minor version must be greater than or equal to the specified version.</td>
</tr>
<tr>
<td>major_only</td>
<td>The major version must match the specified version; the minor version is ignored.</td>
</tr>
<tr>
<td>upto</td>
<td>The major version must be less than or equal to that specified. If the major versions are equal, the minor version must be less than or equal to that specified.</td>
</tr>
</tbody>
</table>

If the -version option is absent, the command shows compatible version numbers.

Usage

The rpcentry unexport operation removes interface and binding information from an entry in the name service. The argument is a list of one or more entry names from which interface and binding information is to be removed.
Privilege Required: You need both r (read) permission and w (write) permission to the CDS object entry (the target name service entry).

Examples

The following example uses the dcecp syntax to unexport the interface and binding information for an interface. The third command entered shows the rpcentry after the unexport operation. The object UUID remains in the rpcentry.

dcecp> rpcentry show ././subsys/applications/bbs_server
{458fffcbe-98c1-11cd-bd93-0000c08ad5f6 1.0
  {ncacn_ip_tcp 130.105.1.227}}
{76030c42-98d5-11cd-88bc-0000c08ad5f6}
dcecp> rpcentry unexport ././subsys/applications/bbs_server \
> -interface {458fffcbe-98c1-11cd-bd93-0000c08ad5f6 1.0}
dcecp> rpcentry show ././subsys/applications/bbs_server
{76030c42-98d5-11cd-88bc-0000c08ad5f6}
dcecp>

Related Information

Commands:

dcecp  rpccp  rpcprofile
endpoint  rpcentry
rpcgroup

A dcecp object that manages an RPC group entry in the DCE Cell Directory Service (CDS).

Format

```
rpcgroup add  rpcgroup_name_list -member member_name_list
rpcgroup create rpcgroup_name_list
rpcgroup delete rpcgroup_name_list
rpcgroup help [operation I -verbose] [-syntax]
rpcgroup import rpcgroup_name_list -interface interface_id [-object object_uuid] [-max integer] [-noupdate]
rpcgroup list rpcgroup_name_list [-member member_name_list] [-noupdate]
rpcgroup operations
rpcgroup remove rpcgroup_name_list -member member_name_list
```

Parameters

<table>
<thead>
<tr>
<th>operation</th>
<th>The name of one specific rpcgroup operation (subcommand) about which you want help information.</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpcgroup_name_list</td>
<td>Specifies a list of one or more names of the rpcgroups being operated on.</td>
</tr>
</tbody>
</table>

Usage

The rpcgroup object represents an RPC group entry in the Cell Directory Service. Each RPC group is named in the DCE name space, therefore, each operation uses one argument, which is a list of names of the group entries to act on. An RPC group is a container of names of either RPC server entries or other RPC groups. There is no data in the group other than the names of the members.
Related Information

Commands:

<table>
<thead>
<tr>
<th>dcecp</th>
<th>rpccp</th>
<th>rpcentry</th>
<th>rpcprofile</th>
</tr>
</thead>
<tbody>
<tr>
<td>endpoint</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
rpcgroup add

rpcgroup add

Adds a member to the specified group entry in the Cell Directory Service.

Format

rpcgroup add rpcgroup_name_list -member member_name_list

Options

-member member_name_list

This required option declares the name of a member being added to the specified group entry. The member_name_list argument is a list of names of one or more members being added to all of the specified groups. (The names need not exist when they are added.) All members are added to all groups.

Usage

The rpcgroup add operation adds a member to the specified group entry in the CDS. If the rpcgroup_name_list entry does not exist, it will be created. The required rpcgroup_name_list argument is a list of one or more full CDS names of the groups to which you want to add members. An empty string is returned upon successful completion. If member_name_list contains the name of an existing member, the name is ignored and no error is generated.

Privilege Required: You need i (insert) permission to the parent directory. You need both r (read) permission and w (write) permission to the CDS object entry (the target group entry).

Examples

The following commands run dcecp and add the member /./LandS/anthro/Cal_host_3 to the group /./LandS/anthro/Calendar_group:

dcecp> rpcgroup add /./LandS/anthro/Calendar_group \
> -member /./LandS/anthro/Cal_host_3

dcecp>

Related Information

Commands:

dcecp rpccp rpcentry rpcprofile endpoint

348 Command Reference
**rpcgroup create**

Creates a new (empty) **rpcgroup** entry in the CDS.

**Format**

`rpcgroup create rpcgroup_name_list`

**Usage**

The **rpcgroup create** operation creates a new (empty) **rpcgroup** entry in the CDS. Because an empty group is the same as an empty RPC entry or RPC profile, calling **rpcgroup create** is the same as calling **rpcentry create** or **rpcprofile create**. The argument is a list of names of RPC groups being created. An empty string is returned upon successful completion. If the RPC group already exists, an error is returned.

**Privilege Required:** You need i (insert) permission to the parent directory.

**Examples**

The following commands run **dcecp** and creates a new group called `./LandS/anthro/Calendar_group`:

```
dcecp> rpcgroup create ./LandS/anthro/Calendar_group
```

dcecp>

**Related Information**

Commands:

```
dcecp endpoint rpccp rpcentry rpcprofile
```
rpcgroup delete

Removes all of the members from the specified group entry in the CDS.

Format

rpcgroup delete rpgroup_name_list

Usage

The rpcgroup delete operation removes all of the members from the specified group entry in the CDS. The argument is a list of names of RPC group entries whose members are being deleted. To remove the entry itself from CDS, you must use rpcentry delete (note that this also removes any profile members from the entry). An empty string is returned upon successful completion. If the RPC group entry does not exist, an error is generated.

Privilege Required: You need w (write) permission to the CDS object entry (the target group entry).

Examples

The following command removes the group members for /.:/LandS/anthro/Calendar_group from the CDS:

dcecp> rpcgroup delete /.:/LandS/anthro/Calendar_group
dcecp>

Then use the rpcentry delete as follows to delete the group entry from CDS:

dcecp> rpcentry delete /.:/LandS/anthro/Calendar_group
dcecp>

Related Information

Commands:

dcecp       rpccp       rpcentry       rpcprofile
endpoint
rpcgroup help

Returns help information about the rpcgroup object and its operations.

Format

rpcgroup help [operation | -verbose] [-syntax]

Options

- -syntax Displays the syntax diagram of each rpcgroup operation.
- -verbose Displays information about the rpcgroup object.

Usage

The rpcgroup help command is used without an argument or option to return brief information about each rpcgroup operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the rpcgroup object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the rpcgroup help command.

Examples

dcecp> rpcgroup help
add Adds members to a list of RPC Groups.
create Creates a list of empty RPC Groups.
delete Deletes all members from a list of RPC groups.
import Returns the bindings from a list of RPC Groups.
list Returns the members of a list of RPC Groups.
remove Removes members from a list of RPC Groups.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp         rpccp         rpcentry         rpcprofile
endpoint
rpcgroup import

Returns a string binding from the specified RPC group.

**Format**

```
rpcgroup import  rpcgroup_name_list -interface interface_id [-object object_uuid] [-max integer] [-noupdate]
```

**Options**

- **-interface interface_id**
  Declares the interface identifier of one RPC interface. The interface identifier has the following form:
  
  `{interface-uuid major-version.minor-version}`
  
  Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:
  
  `-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}`
  
  Leading zeros in version numbers are ignored.

- **-max integer**
  Specifies the maximum number of string bindings to return. A value greater than one returns a list containing up to the number of bindings specified by the value.

- **-noupdate**
  Usually, name service data is cached locally on each machine in a cell. If a name service inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the `-noupdate` option is not specified, `dcecp` goes to a CDS server to retrieve the required data and updates the local CDS cache. Use the `-noupdate` option to avoid taking the time to update local cache when the local cache is up-to-date.

- **-object object_uuid**
  Declares the UUID of one object. The UUID is a hexadecimal string. For example:
  
  `-object 3c6b8f60-5945-11c9-a236-08002b102989`

**Usage**

The `rpcgroup import` operation returns a string binding from the specified RPC group. The argument is a list of names of RPC groups to import from. This command uses the `-interface` and `-object` options to specify matching bindings. Each of these uses only one value, not a list of values. This command also accepts the `-max` option to specify a number of string bindings to return. The order of bindings returned is arbitrary.

**Privilege Required:** You need `r` (read) permission to the specified CDS object entry (the starting name service entry) and to any CDS object entry in the resulting search path.

**Examples**

The following command imports a binding:
RPC Group Import

dcecp> rpcgroup import /./LandS/anthro/Cal_host 3 \
> -interface {ec1eeb60-5943-11c9-a309-0b002b102989 1.1} \
> -object {cbe6eae0-fb6c-11c9-8e0a-0b002b0f4528} \
{ncacn_ip_tcp 130.105.1.227}
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
</tr>
<tr>
<td>rpccp</td>
</tr>
<tr>
<td>rpcentry</td>
</tr>
<tr>
<td>rpcprofile</td>
</tr>
</tbody>
</table>
rpcgroup list

Returns a list of the names of all members of the specified group.

Format

rpcgroup list rpcgroup_name_list [-member member_name_list] [-noupdate]

Options

- **member member_name_list**
  Specifies a list of names of one or more members to be returned from all groups named in the `rpcgroup_name_list` argument. Use this option to check for specific member names. The value of the `member` option is a list of names of RPC entries, RPC groups, or RPC profiles; they are only references stored in the RPC group and do not have to actually exist outside of the group. All members specified are listed from all RPC groups specified in the argument.

- **noupdate**
  Usually, name service data is cached locally on each machine in a cell. If a name service inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the `noupdate` option is not specified, `dcecp` goes to a CDS server to retrieve the required data, and updates the local CDS cache. Use the `noupdate` option to avoid taking the time to update local cache when the local cache is up-to-date.

Usage

The `rpcgroup list` operation returns a list of the names of all members or requested members of the specified group. The returned names are fully qualified names and are returned in an arbitrary order. The argument is a list of names of RPC groups to have their member names returned.

Privilege Required: You need `r` (read) permission to the CDS object entry (the target group entry).

Examples

The following example lists all the members of the group `./:subsys/applications/infobases`, in the order in which they were added to the group:

dcecp> rpcgroup list ./:subsys/applications/infobases
/.../mycell.goodco.com/subsys/applications/video_server
/.../mycell.goodco.com/subsys/applications/bbs_server
/.../mycell.goodco.com/subsys/applications/audio_server1
/.../mycell.goodco.com/subsys/applications/audio_server2
/.../mycell.goodco.com/subsys/applications/clipart_server
/.../mycell.goodco.com/subsys/applications/photo_server1
/.../mycell.goodco.com/subsys/applications/photo_server2
dcecp>

The following example uses the `member` option to list a specific member of the group `./:subsys/applications/infobases`:
**Related Information**

Commands:

- `dcecp`
- `endpoint`
- `rpccp`
- `rpcentry`
- `rpcprofile`
rpcgroup operations

Returns a list of the operations supported by the rpcgroup object.

Format

rpcgroup operations

Usage

The rpcgroup operations command uses no arguments, and returns a list of the available operations for the rpcgroup object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the rpcgroup operations command.

Examples

dcecp> rpcgroup operations
add create delete import list remove help operations
dcecp>

Related Information

Commands:

  dcecp         rpcpp         rpcentry         rpcprofile
  endpoint
**rpcgroup remove**

Removes one or more members from the specified group.

**Format**

```plaintext
rpcgroup remove rpcgroup_name_list -member member_name_list
```

**Options**

- **-member member_name_list**
  
  This required option lets you specify a list of names of one or more members to be removed from all groups named in the `rpcgroup_name_list` argument. The value of the required `-member` option is a list of names of RPC entries, RPC groups, or RPC profiles. They are only references stored in the RPC group and do not have to actually exist outside of the group. All members specified are removed from all RPC groups specified in the argument.

**Usage**

The `rpcgroup remove` operation removes one or more members from the specified group. The argument is a list of names of RPC groups to have members removed from. The value of the required `-member` option is a list of names of RPC entries, RPC groups, or RPC profiles. If a specified member does not exist in an RPC group, an error is returned.

**Privilege Required:** You need **r (read)** permission and **w (write)** permission to the CDS object entry (the target group entry).

**Examples**

The following command removes the member `./.:subsys/applications/video_server` from the `rpcgroup ./.:subsys/applications/infobases`:

```bash
dcecp> rpcgroup remove ./.:subsys/applications/infobases \ 
>   -member ./.../mycell.goodco.com/subsys/applications/video_server
dcecp>
```

**Related Information**

Commands:

- `dcecp`
- `rpccp`
- `rpcentry`
- `rpcprofile`
- `endpoint`
rpcprofile

A dcecp object that manages an RPC profile entry in the DCE Cell Directory Service (CDS).

Format

```
rpcprofile add profile_name_list -member member_name_list
    [-default | -interface interface_id [-priority priority] [-annotation annotation]]
```

```
rpcprofile create profile_name_list
```

```
rpcprofile delete profile_name_list
```

```
rpcprofile help [operation | -verbose] [-syntax]
```

```
rpcprofile import profile_name_list -interface interface_id [-object object_uuid] [-max integer] [-noupdate]
```

```
rpcprofile list profile_name_list [-member member_name_list] [-noupdate]
```

```
rpcprofile operations
```

```
rpcprofile remove profile_name_list
    [-default | -member member_name -interface interface_id | -annotation annotation | -priority priority]
```

```
rpcprofile show profile_name_list
    [-default | -member member-name] [-interface interface_id [-version versions]] [-priority priority] [-annotation annotation] [-noupdate]
```

Parameters

`operation` The name of one specific `rpcprofile` operation (subcommand) about which you want help information.

`profile_name_list` Specifies a list of one or more names of the RPC profile entries being operated on.

Usage

The `rpcprofile` object represents an RPC profile entry in the CDS. Each operation described below uses one argument that names the RPC profiles being acted on. An RPC profile consists of members (also known as elements in other DCE documentation) that can be RPC server entries, RPC groups, or other RPC profiles; therefore, each member of a profile has a name in the DCE name space. Each profile can also have one member, which is the default member of the profile (called the default profile element).

A profile entry contains no attributes, but it does contain information about each member, which is not contained in the member itself. The information stored for each member includes up to four fields of information consisting of interface and version, a member name, a priority (0 through 7), and an annotation. For example:

```
{d46113d0-a848-11cb-b863-08001e046aa5 2.0} /.../mycell.goodco.com/sec 0 rs_bind
```

Various `rpcprofile` object commands have options that correspond to the fields of information contained in profile members. Specifically, the options are `-interface`, `-member`, `-priority`, and `-annotation`.  

358 Command Reference
## Related Information

Commands:

- dcecp
- endpoint
- rpcentry
- rpcgoup
- rpccp
rpcprofile add

Adds a member to the specified profile entry in the CDS.

Format

```
rpcprofile add profile_name_list -member member_name_list
[-default | -interface interface_id [-priority priority] [-annotation annotation]]
```

Options

- **-annotation annotation**
  Defines an annotation string for the profile member. You can include internal spaces in an annotation by enclosing the string in double quotation marks. You cannot specify this option with the **-default** option. The maximum length for the annotation string is 17 characters.

- **-default**
  Performs the **rpcprofile add** operation for the default profile member. When you use the **-default** option, you cannot use any other option except **-member**.

- **-interface interface_id**
  Required when the **-default** option is not used, to declare the interface identifier of an RPC interface. The **rpcprofile add** command operates on only one **interface_id**. You cannot specify this option with the **-default** option.

  The interface identifier has the following form:

  `{interface-uuid major-version.minor-version}`

  Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

  `-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}`

  Leading zeros in version numbers are ignored.

- **-member member_name_list**
  This required option declares the name of a member being added to the specified profile entry. The **member_name_list** argument is a list of names of one or more members being added to all of the specified profiles.

- **-priority priority**
  Defines a search priority for the new profile member. The priority value is in the range 0 to 7 with zero having the highest priority. By default, a non-default element is assigned a priority value of zero. You cannot specify this option with the **-default** option.

Usage

The **rpcprofile add** operation adds a member to the specified profile entry in the CDS. The argument is a list of names of RPC profiles to have members added to. If the **profile_name_list** entry does not exist, it will be created. The value of the required **-member** option is a list of names that are references to an RPC entry, RPC group, or RPC profile (that is, they do not have to actually exist). Accepts the **-interface**, **-priority**, and **-annotation** options with one value (not a list) each. All members are added to each profile mentioned in the argument list. Also accepts a **-default** option to specify that the member being added is the default profile member (other options, except **-member**, are not valid if the **-default** option is used). An empty string is returned upon successful completion.
Privilege Required: You need i (insert) permission to the parent directory. You need both r (read) permission and w (write) permission to the CDS object entry (the target profile entry).

Examples

The following command adds an element to the cell profile, **cell-profile**, in the local cell:

```bash
dcecp> rpcprofile add /.:/cell-profile \
   > -member /.:/Calendar_profile \
   > -interface ec1eeb6wzerodot-5943-11c9-a309-08002b102989,1.1 \
   > -annotation RefersToCalendarGroups
```

The following commands set up a user profile associated with the cell profile as its default element, and add a user-specific element for the Calendar V1.1 interface, as follows:

```bash
dcecp> rpcprofile add /.:/LandS/anthro/molly_o_profile \
   > -member /.:/cell-profile -default
```

```bash
dcecp> rpcprofile add /.:/LandS/anthro/molly_o_profile \
   > -member {/.:/LandS/anthro/Calendar_group} \
   > -interface {ec1eeb6wzerodot-5943-11c9-a309-08002b102989 1.1} \
   > -annotation {Calendar_Version 1.1_Interface}
```

The added profile member contains the global name of the member (specified using its cell-relative name, `/.:/LandS/anthro/Calendar_group`) and the RPC interface identifier for the Calendar Version 1.1 interface.

Related Information

Commands:

```
dcecp    rpcentry    rpcgroup    rpcp
endpoint
```
rpcprofile create

rpcprofile create

Creates a new profile entry in the CDS.

Format

rpcprofile create profile_name_list

Usage

The rpcprofile create operation creates a new (and empty) profile entry in the CDS. Because an empty profile is the same as an empty RPC entry or RPC group, calling rpcprofile create is the same as calling rpcentry create or rpcgroup create. The argument is a list of names of RPC profiles being created. Returns an empty string on success. If the RPC profile already exists, an error is returned.

Privilege Required: You need i (insert) permission to the parent directory. You also need both r (read) permission and w (write) permission to the CDS object entry (the target profile entry).

Examples

dcecp> rpcprofile create ././users/wards_profile
dcecp>

Related Information

Commands:

dcecp            rpcentry            rpcgroup            rpccp
dcecp endpoint
rpcprofile delete

Deletes all of the profile members from the specified profile in the CDS.

Format

rpcprofile delete profile_name_list

Usage

The rpcprofile delete operation deletes all of the profile members from the specified profile entry in the CDS. To remove the entry itself from CDS, you must use rpcentry delete (note that this also removes any group members from the entry). The profile_name_list argument is a list of names of RPC profile entries whose members are being deleted. An empty string is returned upon successful completion.

Privilege Required:  You need w (write) permission to the CDS object entry (the target profile entry).

Examples

The following command removes the profile members for /:/LandS/anthro/molly_o_profile from the CDS:

dcecp> rpcprofile delete /:/LandS/anthro/molly_o_profile
dcecp>

Then use the rpcentry delete as follows to delete the profile entry from CDS:

dcecp> rpcentry delete /:/LandS/anthro/molly_o_profile
dcecp>

Related Information

Commands:

dcecp       rpcentry       rpcgroup       rpccp
endpoint
rpcprofile help

Returns help information about the rpcprofile object and its operations.

Format

rpcprofile help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each rpcprofile operation.
-verbose Displays information about the rpcprofile object.

Usage

The rpcprofile help command is used without an argument or option to return brief information about each rpcprofile operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the rpcprofile object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the rpcprofile help command.

Examples

dcecp> rpcprofile help
add Adds members to a list of RPC Profiles.
create Creates a list of empty RPC Profiles.
delete Deletes all members from a list of RPC Profiles.
import Returns the bindings from a list of RPC Profiles.
list Returns the names of members of a list of RPC Profiles.
remove Removes members from a list of RPC Profiles.
show Returns the attributes of members of a list of RPC Profiles.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp         rpcentry         rpccp
endpoint      rpcgroup
rpcprofile import

Returns a string binding from the specified RPC profile.

Format

```
rpcprofile import profile_name_list -interface interface_id [-object object_uuid] [-max integer] [-noupdate]
```

Options

- **-interface interface_id**
  Declares the interface identifier of an RPC interface. The `rpcprofile import` command operates on only one `interface_id`, not a list.

  The interface identifier has the following form:
  `{interface-uuid major-version.minor-version}`

  Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

  `-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}`

  Leading zeros in version numbers are ignored.

- **-max integer**
  Specifies the maximum number of string bindings to return. A value greater than one returns a list containing up to the number of bindings specified by the value.

- **-noupdate**
  Usually, name service data is cached locally on each machine in a cell. If a name service inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data may not include a recent CDS update. If the `-noupdate` option is not specified, `dcecp` goes to a CDS server to retrieve the required data, and updates the local CDS cache. Use the `-noupdate` option to avoid taking the time to update the local cache when the local cache is up-to-date.

- **-object object_uuid**
  Declares the UUID of an object. The UUID is a hexadecimal string. For example:

  `-object 3c6b8f60-5945-11c9-a236-08002b102989`

Usage

The `rpcprofile import` operation returns a string binding from the specified RPC profile. The argument is a list of names of RPC profiles to import from. This command uses the `-interface` and `-object` options to specify matching bindings. Each of these uses only one value, not a list of values. This command also accepts the `-max` option to specify a number of string bindings to return. If the value is greater than one, then a list of as many matching bindings less than or equal to the value is returned. The order of bindings returned is arbitrary.

**Privilege Required:** You need `r (read)` permission to the specified CDS object entry (the starting name service entry) and to any CDS object entry in the resulting search path.
Examples

The following example imports a binding using the `rpcprofile import` command.

dcecp> rpcprofile import /./users/wards_profile \
> -interface {458fffcbe-98c1-11cd-bd93-0000c000ad7f 1.0} \
> {ncacn_ip_tcp 130.105.1.227}
dcecp>

Related Information

Commands:

dcecp         rpcentry         rpcgroup         rpccp
endpoint
rpcprofile list

Returns a list of the names of all members of the specified profile.

Format

rpcprofile list profile_name_list [-member member_name_list] [-noupdate]

Options

-**member** member_name_list
  This option declares the names of members of the specified profile entry. The member_name_list argument is a list of names of one or more members to be listed.

-**noupdate**
  Usually, name service data is cached locally on each machine in a cell. If a name service inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data may not include a recent CDS update. If the -**noupdate** option is not specified, dcecp goes to a CDS server to retrieve the required data, and updates the local CDS cache. Use the -**noupdate** option to avoid taking the time to update the local cache when the local cache is up-to-date.

Usage

The rpcprofile list operation returns a list of the names of all or requested members of the specified profile. The returned names are fully qualified names and are returned in an arbitrary order. The argument is a list of names of RPC profiles to have their member names returned. The members are concatenated into one list on output.

**Privilege Required:** You need **r (read)** permission to the CDS object entry (the target profile entry).

Examples

The following command lists the cell profile */./cell-profile in the local cell:

dcecp> rpcprofile list */./cell-profile
/.../mycell.goodco.com/sec
/.../mycell.goodco.com/sec-v1
/.../mycell.goodco.com/sec
/.../mycell.goodco.com/sec
/.../mycell.goodco.com/lan-profile
/.../mycell.goodco.com/fs
/.../mycell.goodco.com/subsys/dce/dfs/bak
dcecp>

Related Information

Commands:

dcecp   rpcentry   rpcgroup   rpccp
dcecp   endpoint
rpcprofile operations

Returns a list of the operations supported by the rpcprofile object.

Format

rpcprofile operations

Usage

The rpcprofile operations command uses no arguments, and returns a list of the available operations for the rpcprofile object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required:  No special privileges are needed to use the rpcprofile operations command.

Examples

dcecp> rpcprofile operations
add create delete import list remove show help operations
dcecp>

Related Information

Commands:

dcecp           rpcentry           rpcgroup           rpccp
endpoint
**rpcprofile remove**

Removes one or more members from the specified profile.

**Format**

```
rpcprofile remove profile_name_list
{-default | -member member_name -interface interface_id | -annotation annotation | -priority priority}
```

**Options**

- **-annotation annotation**
  Defines an annotation string for the profile member being removed. You can include internal spaces in an annotation by enclosing the string in quotation marks (or by using other dcecp quoting mechanisms). This option cannot be specified with any other option. The maximum length for the annotation string is 17 characters.

- **-default**
  Performs the `rpcprofile remove` operation on the default profile member. When you use the `-default` option, all of the other options are not allowed.

- **-interface interface_id**
  Declares the interface identifier of an RPC interface. The `rpcprofile remove` command operates on only one `interface_id`. If you use the `-default` option, this option is not allowed. This option requires the `-member` option. No other option can be specified.

  The interface identifier has the following form:

  `interface-uuid major-version.minor-version`

  Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

  `-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}`

  Leading zeros in version numbers are ignored.

- **-member member_name_list**
  Specifies a list of names of one or more members being removed from all profiles named in the `profile_name_list` argument. The value of the `-member` option is a list of names of RPC entries, RPC groups, or RPC profiles. They are only references stored in the RPC profile and do not have to actually exist outside of the profile. All members specified are removed from all RPC profiles specified in the argument. This option requires the `-interface` option. No other option can be specified.

- **-priority priority**
  Defines a search priority for the profile member being removed. The priority value is in the range of 0 to 7 with zero (0) having the highest priority. This option cannot be specified with any other option. By default, a non-default element is assigned a priority of zero (0).

**Usage**

The `rpcprofile remove` operation removes one or more members from the specified profile. The argument is a list of names of RPC profiles to have members removed from. The members being removed are those that match the values given in the following options:

- `-member` and `-interface`
- `-annotation`
rpcprofile remove

-priority
Any members that match these values are removed from all RPC profiles specified in the argument. It also accepts a -default option, in which case, the above options are not valid and the default profile member is removed. An empty string is returned upon successful completion. If a member specified by the -member and -interface options does not exist in an RPC profile, an error is returned.

Privilege Required: You need r (read) permission and w (write) permission to the CDS object entry (the target profile entry).

Examples
The following example removes the member /./subsys/applications/infobases with interface {baf8c319-998f-11cd-ac7b-0000c08adf56 1.0} from the rpcprofile entry /./users/wards_profile:

dcecp> rpcprofile remove /./users/wards_profile \
> -member /./subsys/applications/infobases \
> -interface {baf8c319-998f-11cd-ac7b-0000c08adf56 1.0}
dcecp>

Related Information
Commands:

dcecp     rpcentry     rpcgroup     rpccp
endpoint
rpcprofile show

Returns a list that shows the attributes of all members of one or more profiles.

Format

rpcprofile show profile_name_list
[-default | [-member member-name] [-interface interface_id [-version versions]] [-priority priority]
[-annotation annotation] [-noupdate]

Options

-annotation annotation
Defines an annotation string for the profile member. You can include internal spaces in an annotation by enclosing the string in double quotation marks (or by using other dcecp quoting mechanisms). If you use the -default option, this option cannot be specified. The maximum length for the annotation string is 17 characters.

-default
Performs the rpcprofile show operation on the default profile member. When you use the -default option, no other options except -noupdate are allowed.

-interface interface_id
Declares the interface identifier of an RPC interface. The rpcprofile show command operates on only one interface_id. If you use the -default option, this option cannot be specified.

The interface identifier has the following form:
{interface-uuid major-version.minor-version}

Although the version numbers are optional, the value defaults to 0 if a version number is omitted. The UUID is a hexadecimal string and the version numbers are decimal strings. For example:

-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}

Leading zeros in version numbers are ignored.

-member member_name
This option lets you specify one member name for which to return profile information. If you use the -default option, this option cannot be specified.

-noupdate
Usually, name service data is cached locally on each machine in a cell. If a name service inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data may not include a recent CDS update. If the -noupdate option is not specified, dcecp goes to a CDS server to retrieve the required data, and updates the local CDS cache. Use the -noupdate option to avoid taking the time to update the local cache when the local cache is up-to-date.

-priority priority
Defines a search priority for the profile member you want. The priority value is in the range 0 to 7 with zero having the highest priority. By default, a non-default member is assigned a priority value of zero. If you use the -default option, this option cannot be specified.

-version versions
Specifies interface version numbers being returned with an rpcprofile show operation. This must be used with the -interface option. Specify versions using one of the following values for the -version option:
rpcprofile show

Table 5. Versions for RPCPROFILE

<table>
<thead>
<tr>
<th>Versions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>The interface version is ignored.</td>
</tr>
<tr>
<td>exact</td>
<td>Both the major and minor versions must match the specified versions.</td>
</tr>
<tr>
<td>compatible</td>
<td>The major version must match the specified version, and the minor version must be greater than or equal to the specified version.</td>
</tr>
<tr>
<td>major_only</td>
<td>The major version must match the specified version; the minor version is ignored.</td>
</tr>
<tr>
<td>upto</td>
<td>The major version must be less than or equal to that specified. If the major versions are equal, the minor version must be less than or equal to that specified.</td>
</tr>
</tbody>
</table>

If the -version option is absent, the command shows compatible version numbers.

Usage

The rpcprofile show operation returns a list that shows the attributes of all members of one or more profiles. The argument is a list of names of RPC profiles to have members returned. An attribute list is returned for each member with all of the entered information. It is always in the order:

interface
member
priority
annotation

If any of the items is not given, they are not included in the output; that is, no place holder is included.

Only those members that match the values specified by the following optional options are returned:

-member
-interface
-version
-annotation
-priority

Each option may have only one value; that is, the value may not be a list. The rpcprofile show command also accepts a -default option, in which case, the above options are not valid and the default profile member is returned.

Privilege Required: You need r (read) permission to the CDS object entry (the target profile entry).

Examples

The following example uses no option to show all the members of a profile.

dcecp> rpcprofile show .:/users/wards_profile
{{458ffccb-98c1-11cd-bd93-00000008ad56 1.0} /.../mycell.goodco.com/subsys/applications/infobases 0}
{{00000000-0000-0000-0000-000000000000 0.0} /.../mycell.goodco.com/cell-profile 0}
{{baf8c319-998f-11cd-ac7b-00000008ad56 1.0} /.../mycell.goodco.com/subsys/applications/infobases 0}
dcecp>

The following example uses the -interface option to show a single member of a profile:
dcecp> rpcprofile show /.:users/wards_profile \
> -interface {baf8c319-998f-11cd-ac7b-0000c08adf56 1.0}
> {{baf8c319-998f-11cd-ac7b-0000c08adf56 1.0} /.../mycell.goodco.com/subsys/applications/infobases 0}

dcecp>

**Related Information**

Commands:

```
dcecp  rpcentry  rpcgroup  rpccp
endpoint
```
A dcecp object that manages the security validation service on a host.

**Format**

- `secval activate [host_name_list]`
- `secval deactivate [host_name_list]`
- `secval help [operation | -verbose] [-syntax]`
- `secval operations`
- `secval ping [host_name_list]`
- `secval status [host_name_list]`

**Parameters**

- `host_name_list` A list of one or more names of host systems whose security validation systems you want to act on. This argument has the format of:
  
  `/.../cell_name/hosts/host_name`

- `operation` The name of one specific secval operation (subcommand) about which you want help information.

**Usage**

The secval object represents the security validation service running on a host, as part of the dced server. This service is responsible for maintaining the security credentials of the host machine.

Access to the commands is based on the ACL to the security validation object for a host. This has the format of:

`/.../cell_name/hosts/host_name/config/secval`

**Related Information**

Commands:

- `dcecp` server
secval activate

Activates a security validation service.

Format

secval activate [host_name_list]

Usage

The secval activate operation activates a security validation service. If it is already activated, an error is returned. The optional host_name_list argument is a list of names of host systems whose security validation services you want to activate. An empty string is returned upon successful completion.

Privilege Required: You must have x (execute) permission to the security validation service object.

Examples

dcecp> secval activate

dcecp>

Related Information

Commands:

dcecp server
secval deactivate

Deactivates a security validation service.

Format

secval deactivate [host_name_list]

Usage

The secval deactivate operation deactivates a security validation service. If it is already deactivated, an error is returned. The optional host_name_list argument is a list of names of host systems whose security validation services you want to deactivate. An empty string is returned upon successful completion.

Privilege Required: You must have s (stop) permission to the security validation service object.

Examples

dcecp> secval deactivate
dcecp>

cdecp> secval deactivate

cdecp>

Related Information

Commands:

dcecp server
secval help

Returns help information about the secval object and its operations.

Format

secval help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each secval operation.
-verbose Displays information about the secval object.
-syntax Displays the syntax diagram of each secval operation.

Usage

The secval help command is used without an argument or option to return brief information about each secval operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the secval object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the secval help command.

Examples

dcecp> secval help
activate Enables the secval service.
deactivate Disables the secval service.
ping Contacts the dced secval to validate the DCE security service.
status Returns 1 if secval is enabled, 0 if not.
help Prints a summary of command-line options.
operations Returns the valid operations for command.

dcecp>

Related Information

Commands:

dcecp server
secval operations

Returns a list of the operations supported by the secval object.

Format
secval operations

Usage
The secval operations command uses no arguments and returns a list of the available operations for the secval object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the secval operations command.

Examples
dcecp> secval operations
activate deactivate ping status help operations
dcecp>

Related Information
Commands:
dcecp server
secval ping

Validates the credentials returned by a DCE security service.

Format

secval ping [host_name_list]

Usage

The secval ping operation validates the credentials returned by a DCE security service. Although this routine is expected to be used infrequently, it can be used to verify that secd is trusted. It returns a 1 if the credentials are valid and a 0 if they are not. The optional host_name_list argument is a list of names of host systems whose security validation services you want to validate. If the argument is a list of host names, then a list is returned with a 1 or a 0 for each server.

Privilege Required: No special privileges are needed to use the secval ping command.

Examples

dcecp> secval ping
1
dcecp>

Related Information

Commands:

dcecp

server
secval status

Returns the activation status of the security validation service.

Format

```
secval status [host_name_list]
```

Usage

The `secval status` command returns a 1 if the `secval` service is activated, and a 0 if it is not. The optional `host_name_list` argument is a list of names of host systems whose security validation services you want to check the status of. If the argument is a list, then a list is returned with an integer for each server.

**Privilege Required:** No special privileges are needed to use the `secval status` command.

Examples

```
dcecp> secval status
1
dcecp>
```

Related Information

Commands:

```
dcecp                   server
```
server

A dcecp object that manages DCE application servers.

Format

server catalog [host_name_list ] [-executing] [-simplename]
server create server_name_list {-attribute attribute_list | attribute options}
server delete server_name_list
server disable server_name_list -interface interface_id_list
server enable server_name_list -interface interface_id_list
server help [operation | -verbose] [-syntax]
server modify server_name_list [-change extended_registry_attribute_list | -add extended_registry_attribute_list | -remove extended_registry_attribute_list [-types]]
server operations
server ping server_name_list
server show server_name_list [-executing]
server start server_name_list [-uuid uuid_list]
server stop server_name_list [-method method]

Parameters

host_name_list A list of one or more DCE host names specifying hosts to catalog servers. Host names have the form:
//cell_name/hosts/host_name

operation The name of one specific server operation (subcommand) about which you want help information.

server_name_list A list of one or more names of servers to act on. Each name in this list conforms to the server_name form previously described. Server names have the form:
//cell_name/hosts/host_name/config/service/name

where service is one of the following:
srvrconf Server configuration service
sicvexec Server execution service
service Server service specified by context

The first two replacements for service uniquely identify the correct service as either the configuration service or the execution service. The third is a simpler but ambiguous term; however, the ambiguity can usually be resolved by context. For example, the stop operation only applies to a srvrexec object.

In cases where the service is still unclear, a srvrconf object is assumed.

If the server name is not fully qualified as shown above, then the server name will be interpreted as a simple server name on the local system. This is very useful, but can produce undesirable server names, as in the second example below.

Example 1:
server

dcecp> server create try_tserver -program . . .
A server configuration object is created on the local system with the name of:
/.../local_cell_name/host/local_host_name/config/srvrconf/try_tserver

Example 2:
dcecp> server create /.:/hosts/foster/try_tserver -program . . .
A server configuration object is created on the local system with the name of:
/.../local_cell_name/hosts/local_host_name/config/srvrconf/./.:/hosts/foster/try_tserver

The one exception to this rule is that if config is specified as a directory name in the
server simple name, an error message displays. You can only use config as a
directory in the fully qualified name.
dcecp> server create /.:/hosts/foster/config/try_server
EUVARzerodot5Rzerodot38E DCED name is missing an object name.

Data Structures

interface_id The interface identifier of an RPC interface that can be in dcecp syntax or string
syntax.

The dcecp syntax has the following form:
(interface-uuid,major-version.minor-version)

The string syntax can have no spaces and has the following form:
interface-uuid,major-version.minor-version

Although version numbers are optional, the value defaults to 0 if you omit a version
number. The UUID is a hexadecimal string and the version numbers are decimal
strings. For example:
-interface {ec1eeb60-5943-11c9-a309-08002b102989 3.11}

Leading zeros in version numbers are ignored.

Usage

The server object refers to servers residing on a host. This one object can affect both the running servers
and the configuration information used by dced to start that server. The distinction is usually obvious by
the definition of the operation or by the name given as an argument. When this is not the case, the
ambiguity is resolved by a required option.

Almost all of these commands contact the dced on the target host to perform their operations.
(Exceptions to this are noted below.) The server object has some operations (for example, disable and
enable) that affect endpoints maintained by the dced. However, server object operations do not operate
on endpoints maintained by hosts based on OSF DCE level 1.0, such as MVS/ESA OpenEdition DCE
Release 1.

Note: The commands server disable and server enable, and server stop using -method rpc, are not
supported on z/OS when the target DCE server is running on a z/OS host.

Some commands operate on a single server while other commands operate on more than one server.
See "Parameters" on page 381 for a description of how to specify server names.

Server configuration objects may contain application-specific Extended Registry Attributes (ERAs). Only
the ERAs can be modified after creation, other attributes cannot.
See the [z/OS DCE Administration Guide](#) for more information about ERAs.

### Attributes

**arguments** `string_list`  
The command line arguments passed to the program on start up. Its value is a list of strings, and it cannot be modified after creation.

**directory** `directory_name`  
The working directory that the server is started with. This cannot be modified after creation. In z/OS DCE, if a directory is not specified, it defaults to `/tmp`.

**entryname** `cds_name`  
Specifies the server's RPC entry name. This cannot be modified after creation.

**gid** `group_id`  
The POSIX gid that the server is started with. This cannot be modified after creation.

**keytabs** `keytab_list`  
A list of UUIDs of related keytab objects where the server stores its keys. This cannot be modified after creation.

**program** `program_name`  
The name of the server program being run. Its value is a string, and it cannot be modified after creation. In z/OS DCE, if the program is a script, `#!` must be the first two characters in the file. Then, `/bin/sh` is started to run the script. Enter the `#!` using the same code page that DCEKERN uses at the time the server is started.

**prerequisites** `uuid_list`  
A list of UUIDs of other server configuration objects that represent servers that must be running before this one is started. This cannot be modified after creation.

In z/OS DCE, this information is not used to start the other servers, it is just a note to the administrator.

**principals** `principal_name_list`  
A list of principal names that the server runs as. For example, `secd` runs as three different principals. A fully qualified name is always returned on output. On input, a relative principal name represents a principal in default cell of the dced. This cannot not be modified after creation.

**services** `attribute_list`  
A list where each element is an attribute list of the following attributes, that cannot be modified after creation:

- **annotation** `annotation`  
  A human readable Portable Character Set (PCS) string describing the service. For compatibility with MVS/ESA OpenEdition DCE Release 1 endpoint map annotation strings, this is not an internationalized string.

- **bindings** `protocol_sequence_list`  
  A list of string bindings identifying the service.

- **flags** `flag_name_list`  
  The value is a list of key words to identify flags for the server. Only one is supported:

  - **disabled**  
    The mapping was marked as disabled in the endpoint map.
ifname interface_name
The name of the interface of the service limited to PCS characters
(specified in the interface definition file).

interface interface_id
The interface identifier (UUID and version) of the service (specified in
the interface definition file).

entryname service_name
The name of the service (limited to PCS characters).

objects object_uuid_list
A list of object UUIDs supported by the service.

This cannot be modified after creation:

starton starting_condition_list
This attribute identifies when a server is started. The value is a list of one or more of
the following, which cannot be modified after creation.

auto Start if an RPC serviced by this server is received by dced. Ignored
for those servers that are repositories. This function is not
supported.

boot Start at system startup.

explicit Start if dced receives a command to start the server (that is, the
server start command in dcecp).

failure Start if dced detects a server exit with a non-successful error code.

Specifying a null value to this attribute means the server is not started. An example of
a possible value is:

{starton {boot explicit failure}}

uid user_id
The POSIX uid that the server is started with. This cannot be modified after creation.

uuid uuid
The internal identifier of the object. It can be specified at creation or automatically
generated; but when created, it cannot be modified.

Server configuration objects may also have ERAs attached to them. ERAs may be manipulated by the
modify operation. See the z/OS DCE Administration Guide for more information about server attributes.

Related Information

Commands:

account dcecp registry
server catalog

Returns a list of the names of all server configuration objects on a specified host.

Format

server catalog [host_name_list] [-executing] [-simplename]

Options

-executing Returns the name of all servers known by dced that are currently running on the specified host.

-simplename Returns names that are relative, not fully qualified, removing the
/.../cellname/hosts/hostname/config/service/
part of the name.

Usage

The server catalog operation returns a list of the names of all server configuration objects on a specified host. If called with the optional -executing option, it returns the name of all server execution objects (running servers) known by dced that are currently running on the specified host. If called with no arguments, it returns information about the local host. The optional argument is a list of host names. If more than one is specified, then the information returned is concatenated. The order of information returned is arbitrary. Fully qualified names are returned by default. The -simplename option can be used to return relative names without the
/.../cellname/hosts/hostname/config/service/
added to each name.

Privilege Required: You must have r (read) permission to the applicable container (configuration or execution) objects, /.../cellname/hosts/hostname/config/srvrconf and /.../cellname/hosts/hostname/config/srvrexec

Examples


dcecp> server catalog ./:/hosts/foster
/.../mycell/hosts/foster/config/srvrconf/try_tserver

dcecp>

Related Information

Commands:

account       dcecp       registry
server create

Creates a server configuration object.

Format

```
server create server_name_list {-attribute attribute_list | attribute options}
```

Options

**-attribute attribute_list**

Lets you specify attributes using an attribute list rather than using the individual attribute options such as `-arguments` and `-keytabs`. The format is:

```
-attribute {{attr attr_value} {attr attr_value}}
```

To create a server with Extended Registry Attributes, you must use the following format:

```
-attribute {{fixed-attr fixed-attr_value } ... {attributes {extended-attr extended-attr_value} ...}}
```

The `-attribute` option is intended for use in scripts when pasting in lengthy attribute lists output by previous commands. When specified during creation, the attributes (except for Extended Registry Attributes) cannot be modified.

See ["Attributes" on page 383](#) for descriptions of attributes.

**-attribute_option attribute_value**

Another way to specify attributes is to create individual attribute options by adding a hyphen (`-`) to any attributes listed in the ["Attributes" on page 383](#). This cannot be used for Extended Registry Attributes. The individual attribute options are intended for use with interactive commands.

See ["Attributes" on page 383](#) for descriptions of attributes.

Usage

The `server create` operation creates a server configuration (`srvrconf`) object. The argument is a list of names of server configuration objects being created. An `-attribute` option with an argument list as a value is required to define attributes for the server being created. This also accepts attribute options. An empty string is returned upon successful completion.

If the `uuid` attribute is specified and more than one server object is being created, the specified UUID is used for the first server object and a new UUID is generated for each of the other server objects.

**Privilege Required:** You must have *i* (insert) permission to the server configuration container object, `/.../cellname/hosts/hostname/config/srvrconf`

Examples

1. Use the following example to create a server:
2. To create a server with extended attributes **hostdata/data** and **srvrconf/additional_environ** from the default extended attributes schema, use the following example:

```
dcecp> server create ./hosts/present/config/srvrconf/dceboa1 -attr {{program tserver} \
> {attributes {hostdata/data stringa} \
> {srvrconf/additional_environ stringb} \
> }}
dcecp>
```

3. To create a create a server object with two interfaces for the server object, use the following example:

```
dcecp> server create \
> ./hosts/present/config/srvrconf/dceboa1 \
> -arguments {-princ dceboa1 -boan 1} \
> -program /dce/harmonic/build/ckime/hmdceboa \
> -entryname ./subsys/ibm/harmonic/dceboa1 \
> -keytabs b7f24a76-bc85-11ce-9bdf-1wzerodotwzerodotwzerodot5ac95331 \ 
> -services { \
> {ifname {client_boa}} \
> {annotation {BOA's interface to Clients}} \
> {interface {8dd8f380-2d8c-11ce-ac8c-10005a8a5296 1.0}} \
> {ifname {oi_boa}} \
> {annotation {BOA's interface to OIs}} \
> {interface {97285fc0-2d8c-11ce-bf5c-10005a8a5296 1.0}} } \
> -principals dceboa1 \
> -starton explicit \
> -directory {/dce/harmonic/build/ckime}
dcecp>
```

**Related Information**

Commands:

```
account        dcecp        registry
```
server delete

Deletes a server configuration object.

Format

server delete server_name_list

Usage

The server delete operation deletes a server configuration (srvrconf) object. The argument is a list of names of server configuration objects to be deleted. An empty string is returned upon successful completion. An error is returned if any of the objects do not exist.

Privilege Required: You must have d (delete) permission to the server configuration object, /.../cellname/hosts/hostname/config/srvrconf/name

Examples

dcecp> server delete./hosts/foster/config/srvrconf/try_tserver
dcecp>

Related Information

Commands:

account dcecp registry
server disable

Disables the specified server.

Format

server disable server_name_list -interface interface_id_list

Options

-interface interface_id_list

Specifies a list of one or more RPC interfaces being disabled. The interface identifier can be in string syntax or dcecp syntax.

See "Data Structures" on page 382 for a description of string and dcecp syntaxes.

Usage

Note: The server disable command is not supported on z/OS when targeted at servers running on z/OS. If this command is issued against DCE servers running on z/OS, the following message displays:

EUA07172E DCE interface error.
   DCE status code: 0x113db069 - Not a supported request.

The server disable operation disables the specified server. Communicates with dced and removes the endpoints for all interfaces registered by the server (except the rpc_mgmt interface) from the endpoint map. The argument is a list of names of server execution (srvreexec) objects. This requires the -interface option to specify a list of interfaces being disabled. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission to the server execution object,/.../cellname/hosts/hostname/config/srvreexec/name

Examples

dcecp> server disable /.:/hosts/foster/config/srvreexec/try_tserver \ > -interface {008bebed-c7c1-1ddc-9cb3-0000c0ba4944 1.0}
dcecp>

Related Information

Commands:

account         dcecp         registry
server enable

Enables the specified server.

Format

server enable server_name_list -interface interface_id_list

Options

-interface interface_id_list
  Specifies a list of one or more RPC interfaces to be enabled. The interface identifier can be in string syntax or dcecp syntax.
  See the "Data Structures" on page 382 for a description of string and dcecp syntaxes.

Usage

Note: The server enable command is not supported on z/OS when targeted at servers running on z/OS. If this command is issued against DCE servers running on z/OS, the following message displays:
EUVA07172E DCE interface error.
  DCE status code: 0x113db069 - Not a supported request.

The server enable operation enables the specified server. It communicates with dced and enables any previously disabled endpoint mapping for all interfaces registered by the server in the endpoint map. The argument is a list of names of server execution (srvrexe) objects. Requires the -interface option to specify a list of interfaces being enabled. An empty string is returned upon successful completion.

Privilege Required: You must have w (write) permission to the server execution object, 
/.../cellname/hosts/hostname/config/srvrexe/name

Examples

dcecp> server enable /.:/hosts/foster/config/srvrexe/try_tserver \
> -interface {008bebed-c7c1-1ddc-9cb3-0000c0ba4944 1.0}
dcecp>

Related Information

Commands:

account dcecp registry
server help

Returns help information about the server object and its operations.

Format

server help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each server operation.
-verbose Displays information about the server object.

Usage

The server help command is used without an argument or option to return brief information about each server operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the server object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the server help command.

Examples

dcecp> server help
catalog Returns the list of srvrconf or srvrexec object names.
create Creates a new server configuration (srvrconf) object.
delete Deletes a server configuration (srvrconf) object.
disable Disables interfaces of server (srvrexec).
enable Enables interfaces of server (srvrexec).
modify Modifies the srvrconf object's variable attributes.
ping Pings a server to see if it is receiving requests.
show Returns the attributes of a srvrconf or srvrexec object.
start Starts the specified server.
stop Stops the specified running server.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

account dcecp registry
server modify

Adds fixed attributes and adds or removes Extended Registry Attributes (ERA) and their values from the server configuration object.

Format

```
server modify server_name_list {-change extended_registry_attribute_list | -add extended_registry_attribute_list | -remove extended_registry_attribute_list [-types]}
```

Options

- **-add extended_registry_attribute_list**
  - Lets you add extended registry attributes that may be defined for your environment. You can specify the attributes being removed as a list of one or more ERAs. See the z/OS DCE Administration Guide for more information about extended registry attributes.

- **-change extended_registry_attribute_list**
  - Lets you change the value of extended registry attributes that were created for your environment. You can specify the attributes being changed as a list of one or more ERAs. See the z/OS DCE Administration Guide for more information on extended registry attributes.

- **-remove extended_registry_attribute_list**
  - Lets you remove extended registry attributes that may be defined for your environment. You can specify the attributes being removed as a list of one or more ERAs. See the z/OS DCE Administration Guide for more information about extended registry attributes.

  Refer to the server application documentation for information about any ERAs supported for the server.

- **-types**
  - Specifies that a list of attribute names instead of names and values was given as the value of the **-remove** option, indicating to remove the entire attribute and not just specified values.

Usage

The **server modify** operation changes, adds, or removes ERAs and their values from the server object. The argument is a list of names of server configuration (svrconf) objects being modified. Accepts the **-change** option, which must have an extended attribute list as its value. Attribute options are not supported for this command. The name is always for a server configuration object. You cannot modify a server execution object.

Privilege Required: You must have **w (write)** permission to the server configuration object, 

```
/.../cellname/hosts/hostname/config/svrconf/name
```

Examples

Use the following example to add attributes from a server configuration object.

```
dcecp> server modify tserver -add {hostdata/data {this is the data}}
dcecp>
```

This example assumes that **tserver** was created using the following command:
dcecp> server create tserver -att {program tservprg}

Related Information

Commands:

account  dcecp  registry
server operations

Returns a list of the operations supported by the server object.

Format

server operations

Usage

The server operations command uses no arguments, and returns a list of the available operations for the server object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the server operations command.

Examples

dcecp> server operations
catalog create delete disable enable modify ping show start stop help operations
dcecp>

Related Information

Commands:
account dcecp registry
server ping

Checks whether a server is receiving client requests.

Format

server ping server_name_list

Usage

The server ping operation queries a server to see if it is receiving requests. This operation communicates directly with the server. The argument is a list identifying the servers to ping. server ping returns a list of values, one for each server specified in the argument, in the same order. The value is 1 if the server is listening for RPC requests, or 0 if it is not. Each argument can be in one of the following formats:

- The name of a server entry in the name space being imported from. For example:
  /./hosts/foster/event_server
  When the server name is entered this way, the server must have an object UUID. If the object UUID for the server is nil, then this command returns a status of rpc_s_binding_incomplete and the following error message displays:
  EUVR00625A RPC cannot process API rpc_mgmt_is_server_listening
  because binding is not remote binding.
  DCE status code: 0x16c9a0fb.
  In this case, use the dcecp endpoint show command to get the fully bound string binding for the server, and enter the command again with the fully bound binding.

- A partially bound string binding with an object UUID specified. For example:
  {00337ea9-d979-1dd8-923f-0000c08ad5f6 ncacn_ip_tcp 15.121.12.72}

- A fully bound string binding. For example:
  {ncacn_ip_tcp 15.121.12.72 1075}

Privilege Required: Depending on the individual server, no special privileges are required.

Examples

dcecp> server ping /./hosts/foster/event_server
1

dcecp>

Related Information

Commands:

account   dcecp   registry
server show

Returns information about servers.

Format

server show  server_name_list [-executing]

Options

-executing Returns an attribute list for a running server rather than its associated configuration object.

Format

The server show operation returns a list of both the fixed attributes and extended registry attributes for the server entries specified in the argument. The argument is a list of names of server object entries. If the names are unclear, server configuration (srvrconf) objects are assumed unless the -executing option is present. If the argument is a list, the output is concatenated into a single list in the order specified.

Privilege Required: You must have r (read) permission to the applicable server configuration (srvrconf) or server execution (srvrexec) objects, /.../cellname/hosts/hostname/config/srvrconf and /.../cellname/hosts/hostname/config/srvrexec

Examples

dcecp> server show /.:/hosts/foster/config/srvrconf/try_tserver
{uuid 003b24d2-a196-1df3-915f-0000c0ba4944}
{program tserver}
{arguments /.:/hosts/foster/test_server}
{prerequisites {}}
{keytabs {}}
{entryname /.:/hosts/foster/test_server}
{services
{[ifname {test server}]
 {annotation {dcecp server test program}}
 {interface {008bebed-c7c1-1ddc-9cb3-0000c0ba4944 1.0}}
 {bindings {ncadg_ip_udp 130.105.5.50}}
 {objects 0073f23a-2e1a-1ddd-b73a-0000c0ba4944}
 {flags {}}
 {entryname /.:/hosts/foster/test_server}}
{principals /.../foster_cell/tserver}
{starton boot explicit failure}
{uid 0}
{gid 0}
{dir /opt/tserver}
}
dcecp>

Related Information

Commands:
account dcecp registry
server start

Contacts a **dced** to start a server based on a server configuration object.

**Format**

```
server start server_name_list [-uuid uuid_list]
```

**Options**

- **-uuid uuid_list**  A list of one or more UUIDs that identify the server being started. If the **-uuid** option is used, only one server name can be specified in the **server_name_list**.

**Usage**

The **server start** operation contacts a **dced** to start a server based on a server configuration object. The argument is a list of names of server configuration objects. Returns the UUID of the started server on success. This is the UUID found in the server configuration (**srvrconf**) object for the server.

**Privilege Required:** You must have **x (execute)** permission to the server configuration (**srvrconf**) object, /*/.../cellname/hosts/hostname/config/srvrconf

**Examples**

```
dcecp> server start ././hosts/foster/config/srvrconf/try_tserver
  d90a0374-eb99-11cd-91b1-080009251352
  dcecp>
```

**Related Information**

Commands:

- account
- dcecp
- registry
server stop

Stops the specified running server processes.

Format

server stop server_name_list [-method method]

Options

-method method  Optionally specifies how dced stops the server. The method must be one of the following:

- error       Use a state-preserving mechanism (that is, SIGABRT).
- hard        Use a hard local mechanism (that is, SIGKILL).
- rpc         Have dced use rpc_mgmt_server_stop_listening. This is the default.
- soft        Use a soft local mechanism (that is, SIGTERM).

Usage

Note: The server stop command using the default -method rpc is not supported on z/OS when the target server is running on a z/OS host. If the default is used, the following message displays:

EUVA07172E  DCE interface error.
           DCE status code: 0x113db069 - Not a supported request.

The server stop operation stops the specified running server processes. The argument is a list of names of server execution (srvrevec) objects. An empty string is returned upon successful completion. This command uses an optional -method option to specify how dced stops the server.

The RPC runtime identifies servers not by name, but by interface, object UUID, and endpoints. Note that if you use the rpc method, the command is unable to distinguish between two or more server instances binding without endpoints to the same interface and using the same object UUID. In this case, the command stops a randomly selected server, not necessarily the one named in server_name_list.

Privilege Required: You must have s (stop) permission on the server execution (srvrevec) object.

Examples

dcecp>  server stop ./hosts/foster/config/srvrevec/try_tserver -method soft
dcecp>

Related Information

Commands:

account dcecp registry
**user**

A **dcecp** task object that allows administrators to easily create, delete, and view user information in a DCE cell.

**Format**

user create **user_name_list** -mypwd **password** -password **password** -group **group_name**
-organization **organization_name** [-force] [-attribute **attribute_list**] [other attribute options]

user delete **user_name_list**

user help [**operation** [-verbose] [-syntax]]

user operations

user show **user_name_list**

**Parameters**

**operation**  The name of one specific **user** operation (subcommand) about which you want help information.

**user_name_list**  A list of one or more principal names to operate on. A name can be a fully qualified principal name as in:

*/.../cellname/principal_name*

Or, you can use a cell-relative principal name as in:

**principal_name**

A cell-relative name refers to a principal in the cell named in the _s(sec) convenience variable or in the default cell of the local host if the _s(sec) convenience variable is not set.

Do not use names of registry database objects that contain registry information about principals; principal names must not begin with */./sec/principal/.*

**Usage**

The **user** object represents all of the data associated with a DCE user. This consists of registry information and a CDS directory in the default implementation. The **user** object is a high level representation to allow administrators to easily create, delete, and view user information.

The **user** object makes up a principal and account in the registry, registers the principal in a group and an organization, creating them if necessary, and a CDS directory named after the principal with the appropriate ACL. Only the principal and account attributes are considered attributes of the **user** object, and are the only ones displayed by the **show** operation.

A principal name is used as the name of user objects. The naming conventions for principal names apply; if no cell name is given, it is assumed to be in the current cell. If one is specified, the other cell is contacted.

This object is implemented as a script to allow it to be manipulated and extended on a per-site basis. For example, administrators might want to add GDS and DFS information to the object. Other possible modifications include:

- Changing the location of the CDS directory created for users, or remove it completely.
• Changing the default ACLs placed on the various objects.
• Setting certain attributes or policies on all newly created principals and accounts to match the site policies.
• Setting up site specific defaults for passwords (to be changed by the user later), groups, organizations, and principal directories.
• Supporting a user modify command.

Attributes

client {yes | no}
A flag set to specify whether the account is for a principal that can act as a client. The value of this attribute is either yes or no. If you set this flag to yes, the principal is able to log in to the account and acquire tickets for authentication. The default is yes.

description "text string"
A text string (limited to Portable Character Set (PCS)), usually describes how the user account is used.

dupkey {yes | no}
A flag set to determine if tickets issued to the account principal can have duplicate keys. The value of this attribute is either yes or no. The default is no.

date
The date on which the account expires. To renew the account, change the date in this field. Specify the time using an ISO compliant time format such as:
CCYY-MM-DD-hh:mm:ss
or the string none. The default is none.

forwardabletkt {yes | no}
A flag set to determine whether a new ticket-granting ticket with a network address that differs from the present ticket-granting ticket network address can be issued to the account principal. The proxiabletkt attribute performs the same function for service tickets. The value of this attribute is either yes or no. The default is yes.

fullname string
Used with the create operation to specify the full name of the principal. It is for information purposes only. It usually describes or expands a primary name to allow easy recognition by users. For example, a principal can have a primary name of jsbach and a full name of Johann S. Bach. The value is a string. If it contains spaces, it displays in quotation marks, and must be in quotation marks or braces following dcecp syntax rules when entered. If not entered, the full name defaults to the null string (that is, blank).

group group_name
The name of the group associated with the account. The value is a single group name of an existing group in the registry. This attribute must be specified on the user create command; it does not have a default value. If this group is deleted from the registry, the account is also deleted.

home directory_name
The file system directory in which the principal is placed in at login.

organization organization_name
The name of the organization associated with the account. The value is a single organization name of an existing organization in the registry. This attribute must be specified on the user create command; it does not have a default value. If this organization is deleted from the registry, the account is also deleted.
maxtktlife number_of_hours
The maximum amount of time that a ticket can be valid. Specify the time using the
DTS relative time format ([-]DD-hh:mm:ss). When a client requests a ticket to a server,
the lifetime granted to the ticket takes into account the maxtktlife set for both the
server and the client. In other words, the lifetime cannot exceed the shorter of the
server or the client maxtktlife. If you do not specify a maxtktlife for an account, the
maxtktlife defined as registry authorization policy is used.

maxktktrenew number_of_hours
The amount of time before a principal's ticket-granting ticket expires and that principal
must log in again to the system to re-authenticate and obtain another ticket-granting
ticket. Specify the time using the DTS relative time format ([-]DD-hh:mm:ss). The
lifetime of the principal service ticket can never exceed the lifetime of the principal
ticket-granting ticket. The shorter you make this, the greater the security of the system.
However, because principals must log in again to renew their ticket-granting ticket, the
time needs to consider user convenience and the level of security required. If you do
not specify this for an account, the maxktktrenew lifetime defined as registry
authorization policy is used.

password password
You must create a password for the account. You can use the -password option or
you can create it using the password attribute with the -attribute option and an
attribute_list.

postdatedtkt {yes | no}
A flag set to determine if tickets with a start time some time in the future can be issued
to the account principal. The value for this attribute is either yes or no. The default is
no.

proxiabletkt {yes | no}
A flag set to determine whether a new ticket with a different network address than the
present ticket can be issued to the account principal. The forwardabletkt attribute
performs the same function for ticket-granting tickets. The value for this attribute is
either yes or no. The default is no.

pwdvalid {yes | no}
A flag set to determine whether the current password is valid. If this flag is set to no,
the next time a principal logs in to the account, the system prompts the principal to
change the password. (This flag is separate from the pwdexpdate policy, which sets
time limits on password validity.) The value for this attribute is either yes or no. The
default is yes.

-quota quota
Specifies the total number of registry objects that can be created by the principal. It is
either a non-negative number or the string unlimited. A value of 0 prohibits the
principal from creating any registry objects. Each time a principal creates a registry
object, this value is decremented for that principal.

renewabletkt {yes | no}
A flag set to determine if the ticket-granting ticket issued to the account principal can be
renewed. If this flag is set to yes, the Authentication service renews the ticket-granting
ticket if its lifetime is valid. The value for this attribute is either yes or no. The default
is yes.

server {yes | no}
A flag set to specify whether the account is for a principal that can act as a server. If
the account is for a server that engages in authenticated communications, set this flag
to yes. The value for this attribute is either yes or no. The default is yes.
The path of the shell that is run when a principal logs in.

A flag set to determine whether service tickets issued to the account principal use the standard DCE ticket-granting ticket authentication mechanism. The value for this attribute is either yes or no. The default is yes.

Used with the create operation to specify the User Identifier for the principal. No two principals can have the same uid, however, aliases can share one uid. It is often called the UNIX ID, and it is an integer.

See the [z/OS DCE Administration Guide](https://www.ibm.com/support/knowledgecenter/S57K25_1.2.0/com.ibm.zos.doc/zos_dce administration_guide.html) for more information about attributes.

**Related Information**

**Commands:**

<table>
<thead>
<tr>
<th>account</th>
<th>directory</th>
<th>organization</th>
<th>registry</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>group</td>
<td>principal</td>
<td></td>
</tr>
</tbody>
</table>
user create

Creates a principal name, an account, and a directory in CDS for a single DCE user.

Format

user create user_name_list -mypwd password -password password -group group_name
-organization organization_name [-force] [-attribute attribute_list] [other attribute options]

Options

-attribute attribute_list
   Lets you specify attributes using an attribute list rather than using the individual attribute options such as -group and -organization. The format is:

   -attribute {{attr attr_value} {attr attr_value}}

   You must specify the group, organization, mypwd, and password attributes on the command line. See "Attributes" on page 400 for descriptions of all attributes.

-attribute_option attribute_value
   As an alternative to using the -attribute option with an attribute list, you can create individual attribute options by adding a hyphen (-) before any attributes listed in "Attributes" on page 400. The -attribute option is intended to be used in scripts when pasting in lengthy attribute lists output by previous commands. The individual attribute options might be easier to use for interactive commands.

-force
   Causes the command to create the specified group or organization if they do not already exist.

-group group_name
   You must specify the name of the group associated with the account. The value is a single group name of an existing group in the registry. You can use the -group option or you can create it using the group attribute with the -attribute option and an attribute_list. If this group is deleted from the registry, the account is also deleted.

-mypwd password
   Lets you enter your password. You must enter your password to create an account. This prevents a malicious user from using an existing privileged session to create unauthorized accounts. You can use the -mypwd option or you can create it using the mypwd attribute with the -attribute option and an attribute_list.

-organization organization_name
   You must specify the name of the organization associated with the account. The value is a single organization name of an existing organization in the registry. You can use the -organization option or you can create it using the organization attribute with the -attribute option and an attribute_list. If this organization is deleted from the registry, the account is also deleted.

-password password
   You must create a password for the account. You can use the -password option or you can create it using the password attribute with the -attribute option and an attribute_list.
user create

Usage

The user create operation creates a principal name, an account, and a directory in CDS for one or more DCE users. The argument is the name of new principals being added to the registry. An empty string is returned upon successful completion. On an error, it attempts to undo any interim operations that completed.

For each new user, the command creates a principal and an account for that principal. If either one already exist, an error is generated. The principal is then added to the specified group and organization. Because the principal is created, it cannot already be a member of either the group or organization. If the group or organization does not exist, an error is generated unless the -force option is used to create the group or organization. The command then creates a CDS directory called I:/users/user_name and adds an ACL entry to the default ACL so that the user has rwtci permissions on the directory. This allows all access except for deleting the directory and administering replication on the directory.

Attributes and policies for the newly created principal and account may be specified with the -attribute option and an attribute list as the value, or with attribute options. This command attempts to add any unknown attributes as ERAs on the created principal object. Policies of the organization may not be specified as these may affect more than just the created user. You must specify the group, mypwd, organization, and password attributes on the command line (either in an attribute list or with attribute options). Attributes can be specified in an attribute list or as an attribute option. If specified in both ways, the attribute list value is used.

To protect the account password being entered, enter the account create command only from within the dcecp program. You cannot enter this command from the operating system prompt using dcecp with the -c option.

Privilege Required: Because the user create command performs several operations, you need the permissions associated with each operation:

- To create the principal name, you must have i (insert) permission to the directory in which the principal is being created.
- To add the principal to the group and organization, you must have rM (read, Member_list) permission to the group and to the organization, and rg (read, groups) permission to the principal.
  If the specified groups or organizations do not already exist and you use the -force option, you must have i (insert) permission to the directories in which the groups and organizations are being created.
- To create the account, you must have gmau (groups, mgmt_info, auth_info, user_info) permission to the principal named in the account, rtM (read, text, Member_list) permission to the organization named in the account, tM (test, Member_list) permission to the group named in the account, and r (read) permission on the Registry policy object.
- To create the directory in CDS, you must have the following permissions:
  - ri (read, insert) permissions to the parent directory
  - w (write) permission to the clearinghouse in which the master replica of the new directory is being stored
Examples

The following example creates a principal named **K_Parsons**, adds a group named **users** and an organization named **users**, in a group:

```
dcecp> user create K_Parsons -mypwd osfosf -password change.me \ 
  > -group users -organization users
dcecp> group list users
/.../mycell.goodco.com/W_Rosenberry
/.../mycell.goodco.com/J_Severance
/.../mycell.goodco.com/J_Hunter
/.../mycell.goodco.com/B_Carr
/.../mycell.goodco.com/E_Vliet
/.../mycell.goodco.com/J_Egan
/.../mycell.goodco.com/F_Willison
/.../mycell.goodco.com/K_Parsons

dcecp> account show K_Parsons
{acctvalid yes}
{client yes}
{created /.../mycell.goodco.com/cell_admin 1996-07-27-13:02:51.000+00:00I---
 {description {}}
{dupkey no}
{expdate none}
{forwardabletkt yes}
{goodsince 1994-07-27-13:02:51.000+00:00I-----}
{group users}
{home /}
{lastchange /.../mycell.goodco.com/cell_admin 1994-07-27-13:02:51.000+00:00I
 {organization users}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}
dcecp>
```

Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>directory</th>
<th>organization</th>
<th>principal</th>
<th>registry</th>
</tr>
</thead>
</table>
user delete

Deletes a DCE user.

Format

user delete user_name_list

Usage

The user delete operation deletes one or more DCE users. This deletes the principal from the registry, which also deletes the account and removes the principal from any groups and organizations. This command also deletes the $DCE/l.s/users/user_name directory and any contents. The argument is the principal name of a single DCE user. An empty string is returned upon successful completion.

Privilege Required: Because the user delete command performs several operations, you need the permissions associated with each operation:

- You must have d (delete) permission to the directory in which the target principal exists. You must have rD (read, Delete_object) permission on the principal being deleted.
- You must have rM (read, Member_list) permission on the target groups and organizations and r (read) permission on the member being removed.
- To delete the account, you must have rmau (read, mgmt_info, auth_info, user_info) permissions for the principal named in the account.
- To delete the directory in CDS, you must have d (delete) permission to the directory and w (write) permission to the clearinghouse that stores the master replica of the directory. The server principal needs A (Admin) permission to the parent directory or d (delete) permission to the child pointer that points to the directory you want to delete.

Examples

The following example deletes user K_Parsons from the cell:

dcecp> user delete K_Parsons

dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>directory</th>
<th>organization</th>
<th>registry</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>group</td>
<td>principal</td>
<td></td>
</tr>
</tbody>
</table>
**user help**

Returns help information about the **user** object and its operations.

**Format**

```
user help [operation | -verbose] [-syntax]
```

**Options**

- **-syntax** Displays the syntax diagram of each **user** operation.
- **-verbose** Displays information about the **user** object.

**Usage**

The **user help** command is used without an argument or option to return brief information about each **user** operation.

Use the optional **operation** argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the **-verbose** option for more detailed information about the **user** object itself.

Use the **-syntax** option to display the syntax diagram of each operation or of only the operation specified using the **operation** argument.

**Privilege Required:** No special privileges are needed to use the **user help** command.

**Examples**

```
dcecp> user help
create   Creates a DCE user.
delete   Deletes a DCE user.
show     Shows the attributes of a DCE user.
help     Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>
```

**Related Information**

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>directory</th>
<th>organization</th>
<th>registry</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>group</td>
<td>principal</td>
<td></td>
</tr>
</tbody>
</table>
user operations

Returns a list of the operations supported by the user object.

Format

user operations

Usage

The user operations command uses no arguments, and returns a list of the available operations for the user object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the user operations command.

Examples

dcecp> user operations
create delete show help operations

dcecp>

Related Information

Commands:

account
dcecp
directory
group
organization
principal
registry
user show

Returns the attributes of a DCE user.

Format

user show user_name_list

Usage

The user show operation returns the attributes of one or more DCE users. This includes principal attributes, account attributes, and policies. The information is returned as if the following commands were run in the following order:

principal show
account show -all

Privilege Required: You must have r (read) permission to the principal named in the account.

Examples

The following example shows principal and account information for user K_Parsons:

dcecp> user show K_Parsons

{fullname {}}
{uid 5129}
{uuid 00001409-a943-21cd-be00-0000c08adf56}
{alias no}
{quota unlimited}
{groups users}
{acctvalid yes}
{client yes}
{created /.../mycell.goodco.com/cell_admin 1996-07-27-13:02:51.000+00:00I---
{description {}}
{dupkey no}
{expdate none}
{forwardabletkt yes}
{goodsince 1994-07-27-13:02:51.000+00:00I-----}
{group users}
{home /}
{lastchange /.../mycell.goodco.com/cell_admin 1996-07-27-13:02:51.000+00:00I
{organization users}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgauth yes}
EUVAR046721 No policy.

dcecp>
user show

Related Information

Commands:

- account
- dcecp
directory
group
organization
principal
registry
utc

A dcecp object that lets you add, compare, and convert timestamps in Universal Time Coordinated (UTC) format.

Format

utc add \{absolute_timestamp relative_timestamp | relative_timestamp absolute_timestamp | relative_timestamp relative_timestamp\}

utc compare \{absolute_timestamp absolute_timestamp | relative_timestamp relative_timestamp\} [-noinaccuracy]

utc convert absolute_timestamp [-gmt]

utc help [operation | -verbose] [-syntax]

utc multiply relative_timestamp \{integer | floating_point_factor\}

utc operations

utc subtract \{absolute_timestamp absolute_timestamp | absolute_timestamp relative_timestamp | relative_timestamp relative_timestamp\}

Parameters

absolute_timestamp
An ISO (International Organization for Standardization) compliant time format of the form CCYY-MM-DD-hh:mm:ss.fff[+|]hh:mm:ss.fff. The TDF component [+|]hh:mm, if present, specifies the offset from UTC and implies local system time. The inaccuracy component (Iss.fff), if present, specifies the duration of the time interval that contains the absolute time.

operation
The name of one specific utc operation (subcommand) about which you want help information.

relative_timestamp
A DTS timestamp in the form [-]DD-hh:mm:ss.fff[+|]hh:mm:ss.fff. Relative times often omit fractions of seconds (the leftmost .fff sequence), and usually lack an inaccuracy component (Iss.fff). For example, express a relative time of 21 days, 8 hours, and 15 minutes as 21-08:15:00.

Usage

The utc object lets you operate on timestamps.

Related Information

Commands:
clock dcecp dts utc

Chapter 2. DCE Administration Commands 411
utc add

Adds two timestamps.

Format

utc add (absolute_timestamp relative_timestamp | relative_timestamp absolute_timestamp | relative_timestamp relative_timestamp)

Usage

The utc add command returns the sum of two timestamps. The timestamps can be two relative times or a relative time and an absolute time, in any order.

Privilege Required: No special privileges are needed to use the utc add command.

Examples

dcecp> utc add 1996-05-18-13:21:50.419-04:00I----- +0-00:0-00:02.000I-----
dcecp>

Related Information

Commands:

clock
dcecp
dts
utc
utc compare

Compared two absolute or two relative timestamps indicating the temporal order.

Format

`utc compare {absolute_timestamp absolute_timestamp | relative_timestamp relative_timestamp} [-noinaccuracy]`

Options

- `-noinaccuracy` Specifies to ignore inaccuracies in comparisons. A return of 0 (zero) specifies the times are the same.

Usage

The `utc compare` command uses two timestamps and returns `-1` if the first is earlier, `1` if the second is earlier, and `0` if it cannot be determined. This accepts a `-noinaccuracy` to ignore inaccuracies in comparisons; in this case, a return of `0` (zero) specifies the times are the same.

Privilege Required: No special privileges are needed to use the `utc compare` command.

Examples

```
dcecp> utc compare 1995-10-18-13:22:32.816-04:00I------ \
> 1995-10-18-13:21:50.419-04:00I------ -noinaccuracy
1
dcecp>
```

Related Information

Commands:

- `clock`
- `dcecp`
- `dts`
- `utc`
utc convert

Converts a timestamp from UTC to local time.

Format

`utc convert absolute_timestamp [-gmt]`

Options

- `-gmt` Specifies Greenwich mean time.

Usage

The `utc convert` command accepts a timestamp, returning another timestamp that expresses the same time in the local time zone. If called with the `-gmt`, it returns a Greenwich mean time (GMT) formatted timestamp.

Privilege Required: No special privileges are needed to use the `utc convert` command.

Examples

```
dcecp> utc convert 1995-10-18T13:22:32.816-00:00I------
1995-10-18T09:22:32.816-04:00I------
dcecp>
```

Related Information

Commands:

```
clock       dcecp       dts       utc
```
utc help

Returns help information about the utc object and its operations.

Format

utc help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each utc operation.
-verbose Displays information about the utc object.

Usage

The utc help command is used without an argument or option to return brief information about each utc operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the utc object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the utc help command.

Examples

dcecp> utc help
add Adds a relative and absolute, or two relative, timestamps.
compare Compares two timestamps to decide which is earlier.
convert Converts a timestamp into the local timezone or GMT.
multiply Multiplies a relative timestamp by a number.
subtract Returns the difference between two timestamps.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

clock dcecp dts utc
utc multiply

Multiplying a relative time (a length of time) by an integer or floating point factor.

Format

\texttt{utc multiply \textit{relative\_timestamp} \{\textit{integer} \mid \textit{floating\_point\_factor}\}}

Options

\textit{floating\_point\_factor}

A floating point number such as 53.234.

\textit{integer}

A whole number such as 79.

Usage

The \texttt{utc multiply} command accepts two arguments, a relative timestamp and an integer or a floating point factor. This operation multiplies the length of time (specified with the relative timestamp) by the integer or floating point factor, returning the product as a relative timestamp.

\textbf{Privilege Required:} No special privileges are needed to use the \texttt{utc multiply} command.

Examples

\texttt{dcecp> utc multiply +0-00:00:05.000I----- 3}
\texttt{+0-00:00:15.000I-----}
\texttt{dcecp>}

Related Information

Commands:

\texttt{clock} \hspace{1cm} \texttt{dcecp} \hspace{1cm} \texttt{dts} \hspace{1cm} \texttt{utc}
utc operations

Returns a list of the operations supported by the utc object.

Format

utc operations

Usage

The utc operations command uses no arguments, and returns a list of the available operations for the utc object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the utc operations command.

Examples

dcecp> utc operations
add compare convert multiply subtract help operations
dcecp>

Related Information

Commands:
clock dcecp dts utc
utc subtract

utc subtract

Subtracts one timestamp from another returning the difference as a relative timestamp.

Format

utc subtract {absolute_timestamp absolute_timestamp | absolute_timestamp relative_timestamp | relative_timestamp relative_timestamp}

Usage

The utc subtract command returns the difference between two timestamps that express either an absolute time and a relative time, two relative times, or two absolute times. The return value is a relative timestamp.

Privilege Required: No special privileges are needed to use the utc subtract command.

Examples

dcecp> utc subtract 1995-10-18-13:22:32.816-00:00I---- +0-00:00:15.000I----
1995-10-18-13:22:17.816+00:00I----
dcecp>

Related Information

Commands:
clock                  dcecp                  dts                  utc
uuid

A `dcecp` object that generates and compares UUIDs (universal unique identifiers).

Format

```
uuid compare uuid uuid
uuid create
uuid help [operation | -verbose] [-syntax]
uuid operations
```

Parameters

- `operation` The name of one specific `uuid` operation (subcommand) about which you want help information.
- `uuid` A UUID of the form `069d9fb6-943e-11cd-a35c-0000c08ad5f6`.

Usage

The `uuid` command generates and compares UUIDs. UUIDs uniquely identify DCE entities such as `principals`, RPC entries, and CDS replicas.

Related Information

Commands:

- `dcecp`
uuid compare

UUID compare

Compares two UUIDs.

Format

`uuid compare uuid uuid`

Usage

The `uuid compare` command compares two UUIDs, returning `1` if they are equal, `0` if not. This differs from `string compare` because there are some old string formats of UUIDs that are supported.

Privilege Required: No special privileges are needed to use the `uuid compare` command.

Examples

dcecp> uuid compare 03bb2688-943e-11cd-8bfd-0000c08ad56\  > 069d9f6b-943e-11cd-a35c-0000c08ad56
0

dcecp>

Related Information

Commands:

dcecp
uuid create

Returns a newly generated UUID.

Format

uuid create

Usage

The uuid create command returns a newly generated UUID. It takes no arguments.

Privilege Required: No special privileges are needed to use the uuid create command.

Examples

dcecp> uuid create
03bb2688-943e-11cd-8bfd-0000c08ad56
dcecp>

Related Information

Commands:

dcecp
uuid help

uuid help

Returns help information about the uuid object and its operations.

Format

uuid help [operation | -verbose] [-syntax]

Options

-syntax Displays the syntax diagram of each uuid operation.
-verbose Displays information about the uuid object.

Usage

The uuid help command is used without an argument or option to return brief information about each uuid operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the uuid object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the uuid help command.

Examples

dcecp> uuid help
compare          Compares two UUIDs for equality.
create           Returns a newly generated UUID.
help             Prints a summary of command-line options.
operations       Returns the valid operations for command.
dcecp>

Related Information

Commands:

dcecp
**uuid operations**

Returns a list of the operations supported by the `uuid` object.

**Format**

`uuid operations`

**Usage**

The `uuid operations` command uses no arguments, and returns a list of the available operations for the `uuid` object. The order of the elements is alphabetic, except that `help` and `operations` are listed last.

**Privilege Required:** No special privileges are needed to use the `uuid operations` command.

**Examples**

```
dcecp> uuid operations
compare create help operations
dcecp>
```

**Related Information**

Commands:

`dcecp`
xattrschema

A dcecp object that manages schema information for extended registry attributes.

Format

```
xattrschema catalog schema_name_list [-simplename]
xattrschema create schema_entry_name_list [-attribute attribute_list | attribute options]
xattrschema delete schema_entry_name_list
xattrschema help [operation | -verbose] [-syntax]
xattrschema modify schema_entry_name_list {-change attribute_list | attribute options}
xattrschema operations
xattrschema rename schema_entry_name -to new_schema_relative_name
xattrschema show schema_entry_name_list
```

Parameters

- **operation**: The name of one specific xattrschema operation (subcommand) about which you want help information.

- **schema_entry_name_list**: A list of one or more names of schema object entries to act on. A schema entry name may look like:
  
  `./:/sec/xattrschema/typename` or `./:/hosts/hostname/config/xattrschema/typename`

- **schema_name_list**: A list of one or more names of schema services to act on. A schema name may look like:
  
  `./:/sec/xattrschema` or `./:/hosts/hostname/config/xattrschema`

Usage

The xattrschema object represents the schema information for an extended registry attribute (ERA). Instances of ERAs may be attached as attributes to principals, groups, and organizations, as well as on server configuration and server processing objects supported by dced. Some operations of other dcecp objects can manipulate ERA instances. This object manipulates the definition of the schema of these ERAs.

Each ERA that can be added to an object, has a type, and possibly a value. The definition of the type is stored in the schema. Each ERA type has one ERA schema object associated with it. For ERAs in the registry, these schema objects have the default name `./:/sec/xattrschema/typename`. This name can differ on a per-cell basis. For ERAs in dced, these schema objects have the name `./:/hosts/hostname/config/xattrschema/typename`. The argument to xattrschema commands is a list of names of schema objects of these forms.

Each ERA type is defined to be attachable to a specific object type. The object type is identified by the UUID of the ACL manager for that object type. This ensures that the naming mechanism used to identify the object for purposes of manipulating its ACL is the same as that used to manipulate its ERAs.
Attributes

aclmgr description

A set that lists the ACL managers that support the object types where ERAs of this type can be created. For each ACL manager type, the permissions required for attribute operations are also specified. Each ACL manager is described with a list, the format is:

\{uuid queryset updateset testset deleteset\}

Where the first is the UUID of the ACL manager, and the rest are the sets of permissions (concatenated permission strings as found in an ACL) required to perform each type of operation. The value of this attribute is actually a list of these lists. For example:

\{{8680f026-2642-11cd-9a43-080009251352 r w t D}\}
\{18dbdad2-23df-11cd-82d4-080009251352 r w t mD}\}

You can also use the name of the ACL manager (such as group) in place of the UUID. This attribute can be modified after creation, but only in a limited way. New ACL managers can be added, but existing ones cannot be removed or changed.

annotation PCS string

This is a comment field used to store information about the schema entry. It is a Portable Character Set (PCS) string.

applydefs \{yes | no\}

Specifies that if this ERA does not exist for a given object on an attribute query, the system-defined default value (if any) for this attribute is returned. If set to no, an attribute query returns an attribute instance only if it exists on the object named in the query. The possible values are yes or no.

encoding type

The type of the ERA. This attribute cannot be modified after creation. Valid values are one of:

any

The value of the ERA can take on any encoding. This encoding type is only valid for the definition of an ERA in a schema entry. All instances of an ERA must have an encoding of some other value.

attrset

The value of the ERA is a list of attribute type UUIDs used to retrieve multiple related attributes by specifying a single attribute type on a query.

binding

The value of the ERA contains authentication, authorization and binding information suitable for communicating with a DCE server.

The syntax is a list of two elements. The first element is a list of security information where the first element is the authentication type, either none or dce, followed by information specific for each type. The type none has no further information. The type dce is followed by a principal name, a protection level (one of default, none, connect, call, pkt, pktinteg, cdmf, or pktprivacy), an authentication service (one of default, none, or secret), and an authorization service (one of none, name, or dce). Examples of three security information lists are:

{none}

{dce /./melman default default dce}

{dce /./melman pktprivacy secret dce}
The second element is a list of binding information, where binding information can be string bindings or server entry names. Two examples of binding information are:

{/.:/hosts/hostname/dce-entity
 .:/subsys/dce/sec/master}
{ncadg_ip_udp:130.105.96.3[1234]
 ncadg_ip_udp:130.105.96.6[1234]}

### byte

The value of the ERA is a string of bytes. The byte string is assumed to be a [pickle](https://docs.python.org/3/library/pickle.html) or is otherwise a self-describing type. It is not likely that attributes of this type are entered manually. The format of output is hexadecimal bytes separated by spaces, with 20 bytes per line. For example, suppose the attribute name was `bindata`:

```
{bindata {00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
  10 11 12 13 22 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
  12 11 12 13}}
```

The braces specify that `bindata` has one value. On input, all white spaces are compressed so that users may enter the data as bytes or words or any combination, whichever is more convenient. Therefore, a user can enter the following as input:

```
{bindata {00010203 0405 06070809 0a0b 0c0d0e0f 10111213
  22212223 2425 26272829 2a2b 2c2d2e2f 12111213}}
```

### i18ndata

The value of the ERA is a string of bytes with a tag identifying the (OSF registered) code set used to encode the data. Although it is not likely that administrators will enter attributes of this type manually, the DCE control program does support entering binary data using the following notations:

```
\ddd where \ddd can be one, two, or three octal digits, and \xhh where \xhh can be any number of hexadecimal digits.
```

### integer

The value of the ERA is a signed 32-bit integer.

### printstring

The value of the ERA is a printable IDL character string using PCS.

### stringarray

An array of PCS strings, represented as a Tcl list of strings.

### uuid

The value of the ERA is a UUID.

### void

The ERA has no value. It is simply a marker that is either present or absent.

### intercell value

Specifies the action to take by the Privilege Server when reading ERAs from a foreign cell. Possible values are:

- **accept**: Accepts ERAs from foreign cells. The only check applied is uniqueness if specified by the `unique` attribute.
- **reject**: Discards ERAs from foreign cells.
- **evaluate**: Starts a trigger function to a server that decides whether the ERA is kept, discarded, or mapped to another value.

This attribute is only advisory in z/OS DCE.
multivalued {yes | no}
Specifies that ERAs of this type may be multi-valued (that is, multiple instances of the same attribute type may be attached to a single registry object). The possible values are either yes or no. This attribute cannot be modified after creation.

Specifying no when creating a schema with encoding type attrset or stringarray does not limit the ERA to a single value. Schemas with these encoding types always allow multiple values.

reserved {yes | no}
If set, then this schema entry cannot be deleted through any interface by any user. The possible values are yes or no.

scope
Specifies the name of a security directory or object in the registry. If it is an object, then only that object may have instances of this ERA attached to them. If it is a directory, then only descendants of this directory may have instances of this ERA attached to them. The default is an empty string that does not limit which objects ERAs may be attached to. For example, if this attribute is set to principal/osf/dce, only principals with a prefix of osf/dce in the name may have this type of ERA. This attribute cannot be modified after creation.

This attribute is only advisory in z/OS DCE.

trigtype type
Specifies if there is a trigger and if so, what type it is. The possible values are none, query, or update. If this attribute is anything other than none, then trigbind must be set. This attribute cannot be modified after creation.

Note:
If you receive status code '0x1712215c' when using this option, your security server does not support creating update triggers.

trigbind binding
Contains binding information for the server that will support the trigger operations. This field must be set if trigtype is not none or if intercell is set to evaluate. The value of this attribute is of the format described by the binding encoding.

unique {yes | no}
Specifies that each instance of the ERA must have a unique value within the cell for a particular object type (for instance, principal). The possible values are yes or no. This attribute cannot be modified after creation. This attribute is only advisory in z/OS DCE.

uuid uuid
The internal identifier of the ERA. The value is a UUID. This attribute cannot be modified after creation. If not specified on the create command, a value is generated by the system.

Related Information

Commands:

account
group
organization
server
dcecp
principal
registry

Chapter 2. DCE Administration Commands 427
xattrschema catalog

Returns a list of all the schema entries defined in the service specified in the required argument.

Format

xattrschema catalog schema_name_list [-simplename]

Options

-simplename Returns only the relative name, not the fully qualified name.

Usage

The xattrschema catalog operation returns a list of all the schema entries defined in the service specified by the argument. Returns fully qualified names by default. The -simplename option returns only the relative name, not the fully qualified name.

Privilege Required: You must have r (read) permission to the container object (/./sec/xattrschema or /./hosts/hostname/config/xattrschema).

Examples

dcecp> xattrschema catalog /./sec/xattrschema
/.../mycell/sec/xattrschema/pre_auth_req
/.../mycell/sec/xattrschema/pwd_val_type
/.../mycell/sec/xattrschema/pwd_mgmt_binding
/.../mycell/sec/xattrschema/X500_DN
/.../mycell/sec/xattrschema/X500_DSA_Admin
/.../mycell/sec/xattrschema/disable_time_interval
/.../mycell/sec/xattrschema/max_invalid_attempts
/.../mycell/sec/xattrschema/passwd_override
/.../mycell/sec/xattrschema/test_any
/.../mycell/sec/xattrschema/test_void
/.../mycell/sec/xattrschema/test_printstring
/.../mycell/sec/xattrschema/test_printstring_array
/.../mycell/sec/xattrschema/test_integer
/.../mycell/sec/xattrschema/test_bytes
/.../mycell/sec/xattrschema/test_confidential_bytes
/.../mycell/sec/xattrschema/test_i18n_data
/.../mycell/sec/xattrschema/test_uuid
/.../mycell/sec/xattrschema/test_attr_set
/.../mycell/sec/xattrschema/test_binding dcecp>

dcecp>

Related Information

Commands:

account group principal server
dcecp organization registry
xattrschema create

Creates a new schema entry for an ERA.

Format

```
xattrschema create schema_entry_name_list {-attribute attribute_list \ attribute_options}
```

Options

-attribute attribute_list

Lets you specify attributes using an attribute list. The format is:

-attribute \{(attr value) \{attr value\} ... \{attr value\}\}

See “Attributes” on page 425 for descriptions of the attributes.

-attribute_options attribute_value

As an alternative to using the -attribute option with an attribute list, you can create individual attribute options by adding a hyphen (-) before any attributes listed in the “Attributes” on page 425. The -attribute option is intended to use in scripts when pasting paste in lengthy attribute lists output by previous commands. The individual attribute options may be easier to use for interactive commands.

Usage

The xattrschema create operation creates a new schema entry for an ERA. The argument is a list of one or more names of schema entries being created. Attributes for the created schema entries can be specified using attribute lists (with an -attribute option) and attribute options. If the command argument contains more than one schema name, you cannot specify a UUID attribute. All attributes are applied to all entries being created. An empty string is returned upon successful completion.

Note:

If you receive status code '0x1712215c' when specifying -trigtype update, your security server does not support creating update triggers.

Privilege Required: You must have i (insert) permission to the container object (/./sec/xattrschema or /./hosts/hostname/config/xattrschema).

Examples

```
dcecp> xattrschema create /./sec/xattrschema/test_integer \ 
> -encoding integer -aclmgr {group r r r r}
dcecp>
```

Related Information

Commands:

```
account  group  principal  server
```
```
dcecp  organization  registry
```

Chapter 2. DCE Administration Commands 429
xattrschema delete

 Deletes a schema entry.

 Format

 xattrschema delete schema_entry_name_list

 Usage

 The xattrschema delete operation deletes a schema entry. The argument is a list of names of schema entries being deleted. This command also deletes all ERA instances of the schema entry. If the entries do not exist, an error is generated. An empty string is returned upon successful completion.

 Do not delete a schema that is in use by a server; unpredictable results on the server will occur. Use the server modify command to remove the extended attribute from the server before you delete the schema.

 Privilege Required: You must have d (delete) permission to the container object (/./sec/xattrschema) for security ERAs, or d (delete) permission to the ERA object /./hosts/hostname/config/xattrschema/typename for server ERAs.

 Examples

 dcecp> xattrschema delete /./sec/xattrschema/test_integer
 dcecp>

 Related Information

 Commands:

 account  group  organization  server
dcecp  principal  registry
xattrschema help

Returns help information about the xattrschema object and its operations.

Format

xattrschema help [operation | -verbose] [-syntax]

Options

- syntax Displays the syntax diagram of each xattrschema operation.
- verbose Displays information about the xattrschema object.

Usage

The xattrschema help command is used without an argument or option to return brief information about each xattrschema operation.

Use the optional operation argument for additional information about the options that are valid for that operation. If the operation has no options, an informational message is returned. You can also use the -verbose option for more detailed information about the xattrschema object itself.

Use the -syntax option to display the syntax diagram of each operation or of only the operation specified using the operation argument.

Privilege Required: No special privileges are needed to use the xattrschema help command.

Examples

dcecp> xattrschema help
catalog Returns a list of all entries in a schema.
create Creates a schema entry.
delete Deletes a schema entry.
modify Modifies a schema entry.
rename Renames a schema entry.
show Returns the attributes of a schema entry.
help Prints a summary of command-line options.
operations Returns the valid operations for command.
dcecp>

Related Information

Commands:

account
dcecp

group
organization

principal
registry

server
xattrschema modify

Changes attributes of schema entries.

Format

xattrschema modify schema_entry_name_list { -change attribute_list | attribute options }

Options

-attribute_options attribute_value

As an alternative to using the -change option with an attribute list, you can create individual attribute options by adding a hyphen (-) before any attributes listed in the "Attributes" on page 425. The -change option is intended for use in scripts when pasting in lengthy attribute lists output by previous commands. The individual attribute options may be easier to use for interactive commands.

See "Attributes" on page 425 for a description of the possible attributes.

-change attribute_list
Lets you specify attributes using an attribute list. The format is:

-change {{attr value} {attr value} ... {attr value}}

See "Attributes" on page 425 for descriptions of the attributes.

Format

The xattrschema modify operation changes attributes of schema entries in the security service only. The argument is a list of names of schema entries to be operated on. All modifications are applied to all schema entries named in the argument. Schema entries are modified in the order they are listed and all modifications to an individual schema entry are atomic. Modifications to multiple schema entries are not atomic. A failure for any one schema entry in a list creates an error, and the rest of the operation ends. An empty string is returned upon successful completion.

The -change option can be used to modify the value of any one of the attributes that can be modified. Its value is an attribute list describing the new values for the specified attributes. See "Attributes" on page 425 to determine if the attribute can be modified.

This command also supports attribute options.

Privilege Required: You must have m (mgmt_info) permission to the container object (/:/sec/xattrschema) for security ERAs, or rw (read, write) permissions to the ERA object /:/hosts/hostname/config/xattrschema/typename for server ERAs.

Examples

dcecp> xattrschema modify /:/sec/xattrschema/test_integer \\>
> -aclmgr {organization r r r r}
dcecp>
### Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>group</th>
<th>principal</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>organization</td>
<td>registry</td>
<td></td>
</tr>
</tbody>
</table>
xattrschema operations

Returns a list of the operations supported by the xattrschema object.

Format

xattrschema operations

Usage

The xattrschema operations command uses no arguments, and returns a list of the available operations for the xattrschema object. The order of the elements is alphabetic, except that help and operations are listed last.

Privilege Required: No special privileges are needed to use the xattrschema operations command.

Examples

dcecp> xattrschema operations
catalog create delete modify rename show help operations
dcecp>

Related Information

Commands:

account
group
principal
server
dcecp
organization
registry
xattrschema rename

Changes the name of a schema entry.

Format

xattrschema rename schema_entry_name -to new_schema_relative_name

Options

- **-to new_schema_relative_name**
  Specifies the new name of the schema, in simple name format; that is, without
  \(./sec/xattrschema\) or \(./hosts/hostname/config/xattrschema\)

Usage

This operation changes the name of a specified schema entry. The argument is a single name of an ERA
schema to be renamed. This command uses a required -to option with a value of the new name. The
value may not be a list. An empty string is returned upon successful completion.

Note:

Do not include the path name with the name supplied with the -to option. The path name is only required
when supplying the old schema_entry_name.

Privilege Required: You must have m (mgmt_info) permission to the container object
\(./sec/xattrschema\) for security ERAs, or rw (read, write) permissions to the ERA object
\(./hosts/hostname/config/xattrschema/typename\) for server ERAs.

Examples

```
dcecp> xattrschema rename ./sec/xattrschema/test_integer -to test_int
```

Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>group</th>
<th>principal</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td>organization</td>
<td>registry</td>
<td></td>
</tr>
</tbody>
</table>

Chapter 2. DCE Administration Commands  435
xattrschema show

xattrschema show

Returns an attribute list describing the specified schema entries.

Format

xattrschema show schema_entry_name_list

Usage

The xattrschema show operation returns an attribute list describing the specified schema entries. The argument is a list of names of schema entries being operated on. If more than one schema entry is given, then the attributes are concatenated. Attributes are returned in arbitrary order.

Privilege Required: You must have r (read) permission to the container object (/./sec/xattrschema) for security ERAs, or r (read) permissions to the ERA object /./hosts/hostname/config/xattrschema/typename for server ERAs.

Examples

dcecp> xattrschema show /.:/sec/xattrschema/test_integer
{aclmgr {principal {{query r} {update r} {test r} {delete r}}} }
{annotation {test_integer: encoding type integer}}
{applydefs yes}
{encoding integer}
{intercell reject}
{multivalued yes}
{reserved no}
{scope {}}
{trigbind {none {}}}
{trigtype none}
{unique no}
{uuid 5f439154-2af1-11cd-8ec3-080009353559}
dcecp>

Related Information

Commands:

<table>
<thead>
<tr>
<th>account</th>
<th>group</th>
<th>principal</th>
<th>server</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcecp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>organization</td>
<td>registry</td>
</tr>
</tbody>
</table>
Chapter 3. Remote Procedure Call

This chapter introduces you to the DCE Remote Procedure Call (RPC), describes the RPC control program, and the IDL, and UUIDGEN commands

**Note:** The `rpcd` and `rpccp` facilities are superseded by the DCE Host daemon (`dced`) and the DCE control program (`dcecp`).

It is recommended that `dcecp` be used if the function is available in `dcecp`.

- The DCE daemon is a process that provides the Endpoint Map Service, which maintains the local endpoint map for local RPC servers and looks up endpoints for RPC clients. An endpoint is the address of a specific instance of a server executing in a particular address space on a given system (a server instance). Each endpoint can be used on a system by only one server at a time.

  An endpoint map is a database where servers register their binding information, including endpoints, for each of their RPC interfaces and the associated RPC objects. Each combination of binding information, interface identifier, and object UUID uses a distinct element in the local endpoint map.

- The RPC control program (`rpccp`) provides a set of commands for accessing the operations of the RPC name service interface (NSI). For managing endpoint maps, the control program supports the showing of endpoint map elements and the removing of any set of map elements from the local endpoint map or from any remote endpoint map.

  **Note:** It is recommended that `dcecp` be used if the functionality is available in `dcecp`.

The `rpccp` command starts the RPC control program. All commands, including `rpccp` and its associated commands, are listed alphabetically in this chapter. For information on the logon procedure and other administrative tasks, see the [z/OS DCE Administration Guide](#).

**Exit Values**

The RPC control program reports DCE error messages on the command line. If the command is processed successfully, the internal value returned is 0; otherwise, the value is -1.
add element

Adds an element to a profile in a name service entry; if the specified entry does not exist, creates the entry.

Format

```
rpccp add element profile-entry-name
  -m member
  { -d | -i if-id [-p priority] } [-a annotation] [-s syntax]
```

Options

- **-m**
  Defines a member name for the profile element to be added (required).

- **-d**
  Performs the add element operation on the default profile element. With the -d option, the -i and -p options are ignored.

- **-i**
  Defines an interface identifier for the profile element to be added. Only one interface can be added in a single operation. An interface identifier is required, unless the default profile element is being added. With the -d option, the -i option is ignored.

  The interface identifier has the following form:

  ```
  interface-uuid[.major-version.minor-version]
  ```

  `major-version.minor-version` is optional; defaults to 0,0.

  The UUID is a hexadecimal string, and the version numbers are a decimal string, for example:

  ```
  -i ec1eeb60-5943-11c9-a309-08002b102989,3.11
  ```

  Leading zeros in version numbers are ignored.

- **-p**
  Defines a search priority for the new profile element. The priority value is in the range 0 to 7, with zero having the highest priority. When a default element is added (with the -d option), the -p option is ignored. By default, a non-default element is assigned a priority value of zero.

- **-a**
  Defines an annotation string for the profile element.

  To specify or refer to annotations from within the control program, limit each annotation to an unbroken alphanumeric string, for example, CalendarGroup.

- **-s**
  Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

Parameters

```
profile-entry-name
```

Specifies the entry name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.
**Usage**

The *add element* command adds an element to a profile in a name service entry. The name of the entry containing the profile and the entry name of the profile member in the new element are required. The entry of a profile may have been created previously (by either the *add entry* or *add element* command). However, if the specified entry does not exist, the *add element* command creates the entry.

A profile element (the element added by the *add element* command) is a database record containing the following fields:

- **Interface identifier**
  This is the primary search key. It consists of the interface UUID and the interface version numbers.

- **Member name**
  The entry name of one of the following kinds of RPC name service entries:
  - A server entry for a server offering the requested RPC interface and object
  - A group corresponding to the requested RPC interface
  - A profile.

- **Priority value**
  The priority value, which has 0 as the highest priority and 7 as the lowest, is designated by the creator of a profile element to help determine the order for using the element. NSI search operations randomly select from like priority elements. For the *rpccp add element* command, the default is 0.

- **Annotation string**
  The annotation string enables you to identify the purpose of the profile element. It can be any textual information, for example, an interface name associated with the interface identifier or a description of a service or resource associated with a group. The annotation string is not a search key for the import or lookup operations.

**Privilege Required:** You must have both read permission and write permission to the CDS object entry (the target profile entry). If the entry does not exist, you also require insert permission to the parent directory.

**Note:** The *add element* command is supported in z/OS DCE by the *dcecp rpccprofile add* command.

**Examples**

The following command adds an element to the cell profile, /cell-profile, in the local cell:

```
READY
rpccp
> -i ece1eeb60-5943-11c9-a309-08002b102989,1.1 \
> -m /./:Calendar_profile \
> -a RefersToCalendarGroups \
> /./:cell-profile
```
The following control program commands start the control program, set up a user profile associated with the cell profile as its default element, and add a user-specific element for the Calendar V1.1 interface:

```
add element /.:LandS/anthro/molly_o_profile \
> -d -m /.:cell-profile
add element /.:LandS/anthro/molly_o_profile \
> -m /.:LandS/anthro/Calendar_group \
> -i eclee60-5943-11c9-a309-08002b102909,1.1 \
> -a Calendar_Version_1.1_Interface
```

The added profile element contains the global name of the member (specified using its cell-relative name, /.:LandS/anthro/Calendar_group) and the RPC interface identifier for the Calendar Version 1.1 interface.

**Related Information**

Commands:

- remove element
- remove profile
- show profile
add entry

Adds a name service entry to the name service database.

Format

\texttt{rpccp add entry entry-name [-s syntax]}

Options

- \texttt{-s} Indicates the name syntax of the entry name (optional). The only value for this option is the \texttt{dce} name syntax, which is the default name syntax. Specifying the \texttt{-s} option is unnecessary.

Parameters

\textit{entry-name} Specifies the name of the target name service entry. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The \texttt{add entry} command adds an unspecialized entry to the name service database. The name of the entry is required. The new entry initially contains no NSI attributes, such as binding, object, group, or profile attributes. This command creates a general name service entry for an application or user that can later use the \texttt{export}, \texttt{add element}, and \texttt{add member} commands to make the generic entry into a server entry, a group, or a profile (or a combination), as follows:

- For a server entry, specify the new entry as the target entry for the \texttt{rpccp export} command.
- For a group, specify the new entry as the target group for the \texttt{rpccp add member} command.
- For a profile, specify the new entry as the target profile for the \texttt{rpccp add element} command.

The \texttt{add entry} command enables administrators to add entries for users who lack the required permissions. If you have the permissions required by the \texttt{add entry} command, you can also add an entry using an \texttt{export}, \texttt{add member}, or \texttt{add element} command; if the entry you specify does not exist, the command creates the entry.

\textbf{Privilege Required:} To add an entry, you must have insert permission to the parent directory and both read permission and write permission to the CDS object entry (the target name service entry).

\textbf{Note:} The \texttt{add entry} command is supported in z/OS DCE by the \texttt{dcecp rpcentry create} command.
add entry

Examples

The following commands start RPCCP and add an unspecialized entry to the name service database:

```
READY
  rpccp
rpccp> add entry \
> /.:LandS/anthro/Cal_host_2
```

The following command operates as a TSO command to add an unspecialized entry to the name service database:

```
READY
  rpccp add entry \
> /.:LandS/anthro/Cal_host_3
```

Related Information

Commands:

- add mapping
- remove entry
- show entry
add mapping

Adds or replaces server address information in the local endpoint map.

Format

```
rpccp add mapping [host-address]
   -i if-id
   -b string-binding [ -b...]
   [ -o object-uuid [ -o... ] ]
   [ -a annotation ]
   [ -N ]
```

Options

- **-i** Declares an interface identifier (required). Only one interface can be added in a single operation. The interface identifier has the following form:

  \[interface-uuid[major-version.minor-version]\]

  \[major-version.minor-version\] is optional; defaults to 0,0.

  The UUID is a hexadecimal string, and the version numbers are decimal strings. For example:

  `-i ec1eeb60-5859-11c9-a1c1-8fd12b129891.0`

  Leading zeros in version numbers are ignored.

- **-b** Declares a string binding (required). The string binding must be complete with an endpoint defined. You must also specify an interface identifier (using the `-i` option). The value has the form of an RPC string binding, without an object UUID. For example:


- **-o** Declares an object identifier (optional). Up to 32 objects can be added in a single operation by repeating this option. The object identifier is a hexadecimal string. For example:

  `-o fc1eeb60-5859-11c9-a1c1-8fd12b12989`

- **-a** Declares an annotation string (optional) that is associated to the interface. To specify or refer to annotations from within the control program, limit each annotation to an unbroken alphanumeric string; for example, PhoneDirectory.

- **-N** Declares that the binding specified in this request should not replace existing entries with the same interface and object identifier. A new endpoint map entry will always be added into the endpoint map database.

Parameters

- **host-address** This parameter is a string binding that indicates where to find the target endpoint map. When accessing the local endpoint map, you can specify what protocol sequence to use (optional). For example:

  `ncadg_ip udp:

  When accessing a remote endpoint map, you must specify both a protocol sequence and a network address for the remote system (required). For example:
add mapping


An endpoint is unnecessary in local or remote host addresses, and the remove mapping command ignores any endpoint specified as part of a host address.

Usage

The add mapping command adds or replaces endpoint map entries into the local endpoint map database. The add mapping command enables you to add endpoint mappings into the endpoint map database so that servers need not call rpc_ep_register to register the endpoint with the RPC daemon.

Note: If servers do not call rpc_ep_register to register their endpoints, the servers will not be monitored by the resource manager. Also, the endpoint created by the add mapping command will stay in the DCE host daemon database file, epdb until it is deleted by the remove mapping command until DCEKERN is stopped and restarted, or until the DCE host daemon database file epdb is deleted by the administrator.

This command is usually not used to administer the endpoint map because the server can conveniently perform the same task using the rpc_ep_register application programming interface (API) call directly. This command is provided for convenience only.

Note: The add mapping command is supported in z/OS DCE by the dcecp endpoint create command.

Examples

The following commands start RPCCP and add two endpoint map entries to the endpoint map database (one entry per object):

```
READY
rpccp
rpccp> add mapping \n > -i d5c89800-6dae-11c9-a1c1-08002b102989,0.0 \n > -o fbe696ewzerodot-6dae-11c9-bwzerodot93-wzerodot8wzerodotwzerodot2b1wzerodot2989 \n > -o wzerodot2d52fcwzerodot-6daf-11c9-b958-wzerodot8wzerodotwzerodot2b1wzerodot2989 \n > -b ncadg_ip_udp:9.21.21.46[1232] \n > -a my_interface
```

Related Information

Commands:

remove mapping show mapping
add member

Adds a member to a group in a name service entry; if the specified entry does not exist, creates the entry.

Format

rpccp add member  group-entry-name

-m member

[-s syntax]

Options

-m Declares the name of a member to be added to the specified group entry (required). You can add only one member at a time.

-s Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

Parameters

group-entry-name

Specifies the name of the target group. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The add member command adds a member to a group in a name service entry. The name of the entry containing the group and the name of the new group member are required. The entry of a group may have been created previously (by either the add entry or add member command). If the specified entry does not exist, the add member command creates the entry.

Privilege Required: You must have both read permission and write permission to the CDS object entry (the target group entry). If the entry does not exist, you also require insert permission to the parent directory.

Note: The add member command is supported in z/OS DCE by the dcecp rpcgroup add command.

Examples

The following commands run RPCCP and add the member /./LandS/anthro/Cal_host_3 to the group /./LandS/anthro/Calendar_group:

READY

rpccp

rpccp> add member \
> -m /./LandS/anthro/Cal_host_3 \
> /./LandS/anthro/Calendar_group
add member

Related Information

Commands:

add mapping  remove group  remove member  show group
exit

Causes the DCE RPC control program to end.

Format

rpccp exit

Usage

The exit command causes the rpccp program to end. The original user environment is restored.

Examples

The following example shows the use of the exit command if rpccp was started at the system prompt:

READY
rpccp
rpccp> exit
READY

Related Information

Commands:

quit
Exports binding information for an interface identifier, object UUIDs, or both to a server entry; if the specified entry does not exist, creates the entry.

Format

rpccp export entry-name
   [-i if-id -b string-binding [-b string-binding...]] [-o object-uuid [-o object-uuid...]] |
   [-i if-id -b string-binding [-b...]] |
   [-o object-uuid [-o object-uuid...]] }
   [-s syntax ]

Options

-i Declares the interface identifier of an RPC interface. The export command operates on only one -i option; if you enter more than one, the command ignores all but the last interface identifier. If you specify an interface identifier, you must specify at least one -b option. The -i and -o options can occur together or separately, but one of them is necessary.

The interface identifier has the following form:
interface-uuid,major-version.minor-version

The version numbers are optional, but if you omit a version number, the value defaults to 0. The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:
   -i ec1eeb60-5943-11c9-a390-8002b102989,3.11

Leading zeros in version numbers are ignored.

-b Declares a string binding (optional). To use this option, you must also specify an interface identifier (using the -i option). Each command accepts up to 32 -b options.

The value has the form of an RPC string binding, without an object UUID. The binding information contains an RPC protocol sequence, a network address, and sometimes an endpoint within brackets (rpc-prot-seq:network-addr[endpoint]). For a well-known endpoint, include the endpoint in the string binding, for example:
   -b ncadg_ip_udp:63.0.2.17[5347]

For a dynamic endpoint omit the endpoint from the string binding, for example:
   -b ncadg_ip_udp:16.20.15.25

Note that, depending on your system, string binding delimiters such as brackets ([ ]) may need to be preceded by an escape symbol (\) or placed within quotation marks (" ") at the system prompt. Requirements vary from system to system, and you must conform to the usage rules of your system.

-o Declares the UUID of an object. Each export command accepts up to 32 -o options. The -i and -o options can occur together or separately, but one of them is necessary.

The UUID is a hexadecimal string, for example:
   -o 3c6b8f60-5945-11c9-a236-08002b102989

-s Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.
Parameters

`entry-name` Specifies the name of the target name service entry. Usually, the target is a server entry. However, objects also can be exported (without an interface identifier or any binding information) to a group or a profile.

For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The `export` command places either binding information for an interface identifier, object UUIDs, or both into a server entry, or it places object UUIDs into a group’s entry. The `export` command searches the name service database for the entry with the specified entry name. If it exists, the command uses it; otherwise, the command tries to create a new name service entry using the specified entry name.

Minimally, the command requires the name of the entry and either an interface UUID and binding string or an object UUID.

If the specified entry does not exist, the `export` command tries to create it.

Privilege Required: You must have both read permission and write permission to the CDS object entry (the target name service entry). If the entry does not exist, you also require insert permission to the parent directory.

Note: The `export` command is supported in z/OS DCE by the `dcecp rpccentr export` command.

Examples

This example shows a control program `export` command that is stored in a file for later execution from a JCL prompt. The command exports two objects and an interface with two string bindings to the server entry `/.:/LandS/anthro/Cal_host_3` in the local cell:

```
# file to export Calendar 1.1 at installation time
rpccp export
-i ec1eeb60-5943-11c9-a309-08002b102989,1.1 \
-b ncadg_ip_udp:16.20.15.25 \
-b ncadg_ip_udp:63.0.2.17 \
-o 30dbeee0-fb6c-11c9-8ee8-08002b0f4528 \
-o 16977538-e257-11c9-8dc0-08002b0f4528 \
/.:/LandS/anthro/Cal_host_3
```

The `rpccp` command starts the control program, and the `export` command exports the interface and two string bindings to the server entry `/.:/LandS/anthro/Cal_host_2` in the local cell as follows:

```
rpccp
rpccp> export -i ec1eeb60-5943-11c9-a309-08002b102989,1.1 \
> -b ncadg_ip_udp:16.20.15.25 \
> -b ncadg_ip_udp:63.0.2.17 \
> /.:/LandS/anthro/Cal_host_2
```

The following example shows how object UUIDs can be exported:

```
READY
rpccp
rpccp> export -o 30dbeee0-fb6c-11c9-8ee8-08002b0f4528 \
    -o 16977538-e257-11c9-8dc0-08002b0f4528 \
> /.:/LandS/anthro/Cal_host_2
```
export

Related Information

Commands:

import  show server  unexport
help

Displays a list of commands or the options of a specified command.

Format

rpccp help [rpccp-command]

Parameters

rpccp-command Specifies one of the following control commands:

<table>
<thead>
<tr>
<th>Table 6. RPC Control Program Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>add element</td>
</tr>
<tr>
<td>add member</td>
</tr>
<tr>
<td>help</td>
</tr>
<tr>
<td>query epdb</td>
</tr>
<tr>
<td>rebuild epdb</td>
</tr>
<tr>
<td>remove mapping</td>
</tr>
<tr>
<td>show entry</td>
</tr>
<tr>
<td>show profile</td>
</tr>
</tbody>
</table>

Usage

The help command displays information about the rpccp command set or the options and argument associated with a specific command.

Note: The help command is supported in z/OS DCE by the dcecp help command.

Examples

The following command operates from the system prompt to display internal commands of the control program:
READY

**rpccp help**

Usage: rpccp subcommand {entry-name|host-address} [option [value]] ...

**subcommands:** quit (or exit)
help
show entry -isu
add entry -s
remove entry -s
show group -mrsu
remove group -s
add member -ms
remove member -ms
show profile -adimrsuv
remove profile -s
add element -adimps
remove element -adims
show server -iosu
export -bios
unexport -iosv
import -einosu
show mapping -iov
remove mapping -bio
add mapping -abioN
rebuild epdb
query epdb
ping -w

rpccp>

The following command starts the control program and displays the syntax of the *remove entry* command:

READY

**rpccp**

rpccp> help remove entry

Usage: rpccp remove entry entry-name [option value]

options:  s  syntax  name  syntax

**Related Information**

Commands:

| add element | exit | remove group | show group |
| add entry | export | remove mapping | show mapping |
| add mapping | import | remove member | show profile |
| add member | quit | remove profile | show server |
| epdb query | remove element | rpccp | unexport |
| epdb rebuild | remove entry | show entry |
Calls the Interface Definition Language (IDL) compiler, either from the OMVS shell or from TSO/E. By default, starts the C compiler unless -keep is specified.

Format

idl file-name [option] ...

Parameters

file-name The file-name, and any file name used in the options, can be either a hierarchical file system (HFS) or a partitioned data set (PDS) name. If a PDS name is specified, it must have a prefix of //.

Options

-client file_type Determines which client files to generate. If you do not specify this option, the compiler generates all client files. File types are:

none Does not generate client files.
stub Generates only a client stub file.
aux Generates only a client auxiliary file. A client auxiliary file is generated only if the interface contains any out-of-line or self-pointing types.

See the Important Note to Users under "Usage" on page 458.

all Generates client stub and client auxiliary files. This is the default and is the same as not specifying the -client option.

See the Important Note to Users under "Usage" on page 458.

-server file_type Determines which server files to generate. If you do not specify this option, the compiler generates all server files. File types are:

none Does not generate server files.
stub Generates only a server stub file.
aux Generates only a server auxiliary file. A server auxiliary file is generated only if the interface contains any out-of-line, self-pointing, or pipe types.

See the Important Note to Users under "Usage" on page 458.

-cstub file name Lets you specify a name for the generated client stub file.

Using TSO/E

By default, the compiler takes the base name of the IDL file and appends CS to create a name for the generated client stub file. This file is stored in the DD:USERC data set. If the -cstub option is used, the specified file name becomes the name of the generated client stub file.

Using the Shell

By default, if the generated client stub file is being saved to HFS, the compiler takes the base name of the IDL file and appends _cstub.c to create a name for the generated client stub file. If you use the -cstub option, the IDL compiler takes the specified file name and appends .c to create a name for the generated client stub file.
If you save the output to a PDS using the `-out` option without the `-cstub` option, the IDL compiler takes the base name of the IDL file and appends `CS` to create a name for the generated client stub file. If the `-cstub` option is used, the specified file name becomes the name of the generated client stub file.

**-sstub file name** Lets you specify a name for the generated server stub file.

**Using TSO/E**

By default, the compiler takes the base name of the IDL file and appends `SS` to create a name for the generated server stub file. This file is stored in the `DD:USERC` data set. If the `-sstub` option is used, the specified file name becomes the name of the generated server stub file.

**Using the Shell**

By default, if the generated server stub file is being saved to HFS, the compiler takes the base name of the IDL file and appends `_sstub.c` to create a name for the generated server stub file. If you use the `-sstub` option, the IDL compiler creates a name for the generated server stub file by appending `.c` to the specified file name.

If you save the output to a PDS using the `-out` option, without the `-sstub` option, the IDL compiler takes the base name of the IDL file and appends `SS` to create a name for the generated server stub file. If the `-sstub` option is used, the specified file name becomes the name of the generated server stub file.

**-caux file name** Lets you specify a name for the generated client auxiliary file.

See the Important Note to Users under "Usage" on page 458.

**Using TSO/E**

By default, the compiler takes the base name of the IDL file and appends `CA` to create a name for the generated client auxiliary file. This file is stored in the `DD:USERC` data set. If the `-caux` option is used, the specified file name becomes the name of the generated client auxiliary file.

**Using the Shell**

By default, if the generated client auxiliary file is being saved to HFS, the compiler takes the base name of the IDL file and appends `_caux.c` to create a name for the generated client auxiliary file. If you use the `-caux` option the IDL compiler takes the specified file name and appends `.c` to create a name for the generated client auxiliary file.

If you save the output to a PDS using the `-out` option, without the `-caux` option the IDL compiler takes the base name of the IDL file and appends `CA` to create a name for the generated client auxiliary file. If the `-caux` option is used, the specified file name becomes the name of the generated client auxiliary file.

**-saux file name** Lets you specify a name for the generated server auxiliary file.

See the Important Note to Users under "Usage" on page 458.

**Using TSO/E**

By default, the compiler takes the base name of the IDL file and appends `SA` to create a name for the generated server auxiliary file. This file is stored in the `DD:USERC` data set. If the `-saux` option is used, the specified file name becomes the name of the generated server auxiliary file.

**Using the Shell**

By default, if the generated server auxiliary file is being saved to HFS, the compiler takes the base name of the IDL file and appends `_saux.c` to create a name for the generated server auxiliary file. If you use the `-saux` option the IDL compiler takes the specified file name and appends `.c` to create a name for the generated server auxiliary file.
If you save the output to a PDS using the -out option, without the -saux option the compiler takes the base name of the IDL file and appends SA to create a name for the generated server auxiliary file. If the -saux option is used, the specified name is used as the name for the generated server auxiliary file.

-**header** header_file
   Lets you specify a name for the generated header file.

   **Using TSO/E**
   By default, the compiler uses the base name of the IDL file as the name for the generated header file which is stored in the DD:USERH data set. If the -header option is used, the specified file name becomes the name of the generated header file.

   **Using the Shell**
   By default, if the generated header file is being saved to HFS, the compiler takes the base name of the IDL file and appends .h to create a name for the generated header file. If you use the -header option the IDL compiler takes the specified file name and appends .h to create a name for the generated header file.

   If you save the output to a PDS using the -out option, without the -header option the compiler uses the base name of the IDL file as the name for the generated header file. If the -header option is used, the specified name is used as the name for the generated header file.

-**out** directory
   Places the output names in the HFS directory or PDS that you specify.

   **Using TSO/E**
   This option is replaced by ddnames.

   **Using the Shell**
   By default, the compiler places the output files in the current working directory.

   The following example compiles a file test.idl under HFS and saves the output to a PDS called DCE.OUT.STUB.

   ```
   idl -no_cpp -keep c_source -I export test.idl -out "//'DCE.OUT.STUB'"
   ```

-**I** directory
   Specifies a directory name or PDS that contains imported interface definition files or attribute configuration files.

   **Using TSO/E**
   You can specify more than one directory by specifying additional -I DD:ddname options on the TSO/E command line in the JCL. The compiler searches the data sets in the order you list them. If a file is present in more than one data set, the compiler takes the first occurrence. The default behavior of the compiler is to first search the DD:USERIDL data set, then all data sets you specify, and then the DD:SYSSID data set.

   **Using the Shell**
   You can specify more than one directory by specifying additional -I directory options on the command line. A PDS can also be specified using the -I option on the command line.

   The compiler searches the directories in the order you list them. If a file is present in more than one directory, the compiler takes the first occurrence of the file. The default behavior of the compiler is to first search the current directory, then all directories you specify, then the system IDL directory, and finally (for the .acf file only, not for imported .idl files and their corresponding .acf files) the directory in which the .idl file resides when different from the current directory.

   The IDL compiler ignores any PDS with a name ending in .ACF when looking for an IDL file to import.
When looking for an ACF, the IDL compiler searches the -I HFS directories or PDS in the order in which they are specified on the command line. All HFS directories specified are searched; however, only PDS names ending with .ACF are searched.

-no_def_idir Specifies where the compiler is to search for imported files.

Using TSO/E
Specifies that the compiler search only the DD:USERIDL data set for imported files. When you use this with -I DD:ddname, the compiler searches only the data sets you list, not the DD:USERIDL data set nor the DD:SYSIDL data set.

Using the Shell
Specifies that the compiler search only the current directory for imported files. When you use this with -I directory, the compiler searches only the directories you list, not the current directory, and not the system IDL directory.

-no_mepv Causes the compiler not to generate a manager Entry Point Vector (EPV) in the server stub. Use this option if the manager code and IDL file do not use the same operation names. If you specify this option, you must provide an EPV within the manager code that can be used when the interface is registered with the RPC server runtime. The EPV type is if-name_vmajor-version_minor-version_epv_t where if-name is the interface name. It is not necessary to use this option if the operation names in the manager code and IDL file are the same. In this case, the compiler generates a manager EPV in the server stub using the names of the operations in the IDL file. The name of the EPV is if-name_vmajor-version_minor-version_epv where if-name is the interface name. For information on registering the server, see rpc_server_register_if in the z/OS DCE Application Development Guide: Core Components.

-cepv Generates local routines in the client stub file and defines a Client Entry Point Vector (CEPV) of the name if-name_vmajor-version_minor-version_c_epv where if-name is the interface name. The CEPV contains the addresses of the local routines. The client code must call the routines indirectly by using the addresses in the CEPV; otherwise, the stub routines in the client stub file must have the same names as the operations in the IDL file. For information on registering the server, see rpc_server_register_if in the z/OS DCE Application Development Guide: Core Components.

cpp_cmd 'c_preprocessor_command_line'
Specifies a C preprocessor other than the default. This option is not supported by the z/OS DCE IDL compiler.

cpp_opt 'command_options'
Specifies additional options to be passed to the C preprocessor. This option is not supported by the z/OS DCE IDL compiler.

-no_cpp Prevents the call to the C preprocessor for .idl and .acf files. Note that the C preprocessor must be run on files that contain preprocessor directives, such as #include, in the interface definition.

-cc_cmd 'command_line'
Runs the C compiler and the compiler options that you specify in the 'command_line' option rather than the default C compiler and compiler options. This option is not supported by the z/OS DCE IDL compiler.

-cc_opt 'command_options'
Specifies additional options to be passed to the C compiler. This procedure is done for each generated stub or auxiliary file and can only be issued once.

Using TSO/E
You can add options to the TSO/E command line used to run the C compiler. The IDL compiler concatenates the -cc_opt and -D arguments that run the C compiler.
Using the Shell
You can add options to the command line used to run the C compiler. The IDL compiler concatenates the -cc_opt and -I arguments and the source file name into a command that calls the compiler.

-Dname[=definition]
Defines a symbol name and an optional value to be passed to the C preprocessor. You can use this method of defining a symbol instead of using #define in the source code. You can use more than one -Dname option on the command line.

-Uname
Undefines (removes) any initial definition of a symbol name as defined by -Dname. This option is not supported by the z/OS DCE IDL compiler.

-space_opt
Generates code for the marshalling and unmarshalling of data that is optimized for space, rather than for speed. This option is not supported by the z/OS DCE IDL compiler.

-syntax_only
Checks only the syntax the IDL file but does not generate any output files. This option is not supported by the z/OS DCE IDL compiler.

-keep file_types
Specifies which files to keep and can be used to prevent the C compiler from being called (which is the default, if -keep is not used). To produce the object modules, the IDL compiler first creates C source modules, then calls the target C compiler to produce object modules, and, finally, deletes the C source modules. If you do not use -keep, only the object modules are saved. File types are:

none Does not save the C source or the object modules. Does not run the C compiler.
c_source Saves only the C source modules. Does not run the C compiler.
object The default, if -keep was not specified. The C compiler is called and only the object modules are saved.
all The C compiler is called and both the C source and the object module are saved.

-bug n, -no_bug n
Retains (-bug) or does not retain (-no_bug) a specified bug from earlier IDL compiler versions. This option is not supported by the z/OS DCE IDL compiler.

-stdin
Takes the standard output of a previous utility as the input to the IDL command. This option is not supported by the z/OS DCE IDL compiler.

-version
Displays an information message for the z/OS DCE IDL compiler.

-v
Prints informational messages (verbose mode) on the screen while the compiler is running.

-no_warn
Suppresses compiler warning messages.

-confirm
Displays all the idl command options but does not compile the source IDL file. If you use this with the -v option, informational messages about how the compiler behaves if you do not use -confirm are displayed, but no corresponding actions are performed.

-userpfx USERPRFX
Specifies the high level qualifier of the IDL source data set. This parameter is only available in the TSO/E environment.

-c_idl file_name
Lets you specify a name for the generated cidl file.
Using TSO/E
By default, the compiler takes the base name of the IDL file to create a name for the generated cidl file. This file is stored in the DD:USERCIDL data set. If the -cidl option is used, the specified file name becomes the name of the generated cidl file.

Using the Shell
By default, if the generated cidl stub file is being saved to HFS, the IDL compiler takes the base name of the IDL file and appends .cidl to create a name for the generated cidl file. If you use the -cidl option, the IDL compiler takes the specified file name and appends .cidl to create a name for the generated cidl file.

Note: This option is valid only with DCE Application Support.

-no_cidl file_name
Lets you prevent the generation of a cidl file. The -no_cidl option overrides the -cidl option when both are specified.

Note: This option is valid only with DCE Application Support.

-char_is_unsigned_char
Allows the IDL compiler to conform to the original behavior of treating both char and unsigned char data types as unsigned char.

If this option is not specified:

- The IDL compiler treats these as different data types to conform to the type checking of C/C++ compilers. When a variable within an IDL file is defined as an unsigned char, the idl_char type macro is used within the generated H file and within the generated client and server stub files. When a variable within an IDL file is defined as char, the idl_norm_char macro is used.

If this option is specified:

- the idl_char type macro is used for both char and unsigned char.

The idl_char type macro is type defined as unsigned char. The idl_norm_char macro is type defined as char.

This option is typically used when an existing server or client application defines a variable as idl_char and the corresponding IDL file defines the variable as char.

For more information, see the section on Base Type Specifiers in the z/OS DCE Application Development Guide: Core Components.

Usage
The idl command calls the IDL compiler, from the OMVS shell, from TSO/E, or from Batch, to convert an interface definition, written in IDL, into output files. The IDL compiler can access both HFS and PDS files when called from the shell, and only PDS files when called from TSO/E or Batch. If a PDS file is specified, it must have a prefix of //.

The output files include a header file, server stub file, client stub file, and auxiliary files. The compiler constructs the names of the output files by appending to the base name of the interface definition source file name the new suffix appropriate to the newly generated type of output file. For example in TSO/E, DD:USERIDL(MATH) could produce DD:USERC(MATHSS) or DD:USEROBJ(MATHSS) for the server stub.

The IDL compiler starts the C preprocessor with the following options. All other options used are the installation default. If no installation default was created, the IBM default is used.

LONGNAME, PPONLY(80), FLAGS(S), LANGVL(SAAL2)
These options are documented in the *z/OS C/C++ User's Guide* (SC09-4767). These options are important for understanding which columns the IDL compiler accepts for input. If these options are not understood, use only columns 2 through 72 for input in an IDL file.

**Note:** References to the C compiler and C preprocessor refer to IBM C/C++ for z/OS.

---

**Important Note to Users**

The z/OS DCE for IDL compiler does not need aux files. These files are not automatically generated. Makefiles from MVS/ESA OpenEdition DCE Release 1 that contain dependencies on aux files should be updated to remove these dependencies. If this is not feasible, the z/OS IDL compiler will automatically generate dummy auxiliary files if environment variable IDL_GEN_AUX_FILES is exported.

The `idl` command accepts the following input:

- An interface definition file name. In a PDS this file name can be no more than six characters in length, as two characters are reserved for the suffix that IDL appends. There is no restriction on the length of an HFS file name.

When outputting for PDS or HFS, IDL appends the following characters:

<table>
<thead>
<tr>
<th>Table 7. IDL Appended Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PDS</strong></td>
</tr>
<tr>
<td>CS</td>
</tr>
<tr>
<td>SS</td>
</tr>
<tr>
<td>CS</td>
</tr>
<tr>
<td>SA</td>
</tr>
</tbody>
</table>

- Options to indicate either special actions to be performed by the IDL or C compilers, or special properties of the input or output files.

To compile IDL generated stubs, the following C compiler options must be specified:

- In all three environments (the z/OS UNIX System Services shell, TSO/E, and batch), the following definitions are required:
  - `z/OS`
  - `OPEN_SYS`
  - `DCE_THREADS`
- In all three environments, the following compiler option is recommended when compiling C source:
  - `-DLL`
- In the TSO and batch environments, the following C compiler options are also required:
  - `LONGNAME`
  - `NOMAR`
  - `NOSEQ`

**Cautions**

The IDL compiler generates ANSI C code. It also supports C compilers that are not fully ANSI-compliant; a warning message may occur during compilation of the stubs by the C compiler. A C compiler that is not fully ANSI-compliant may generate the following warning messages:
Restrictions

- PDS IDL member (file) names can be no more than eight characters in length.

  Some .idl and .h files shipped with z/OS DCE have names longer than this. Versions of these files with shorter names are shipped for compiling using TSO and batch. You must use these names when compiling under TSO or batch.

<table>
<thead>
<tr>
<th>TSO and Batch Short Name</th>
<th>UNIX Long Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>secatool.h</td>
<td>sec_attr_tools.h</td>
</tr>
<tr>
<td>secabase.idl</td>
<td>sec_attr_base.idl</td>
</tr>
<tr>
<td>secabase.h</td>
<td>sec_attr_base.h</td>
</tr>
<tr>
<td>secatrig.idl</td>
<td>sec_attr_trig.idl</td>
</tr>
<tr>
<td>secatrig.h</td>
<td>sec_attr_trig.h</td>
</tr>
<tr>
<td>secattr.idl</td>
<td>sec_rgy_attr.idl</td>
</tr>
<tr>
<td>secattr.h</td>
<td>sec_rgy_attr.h</td>
</tr>
<tr>
<td>secatts.idl</td>
<td>sec_rgy_attr_sch.idl</td>
</tr>
<tr>
<td>secatts.h</td>
<td>sec_rgy_attr_sch.h</td>
</tr>
</tbody>
</table>

When compiling in the OMVS shell, you can use either the short- or long-named versions of the files, but use one or the other consistently; do not mix them.

- The .c, .h, and .o stub files generated by the IDL compiler are not portable to other DCE platforms. Stub files used on the z/OS DCE platform must be generated with the z/OS DCE IDL compiler.

- The following file names are reserved by the IDL compiler. Do not name an IDL file with one of these names as it may cause unpredictable behavior.

  In an HFS
  - iovector.idl
  - ibase.idl
  - nbase.idl
  - ncastat.idl
  - ndrold.idl
  - rpc.idl
  - rpcbase.idl
  - rpcpvt.idl
  - rpcsts.idl
  - rpcbase.idl
  - rpcpvt.idl
  - rpcsts.idl
  - rpcbase.idl
  - rpcpvt.idl
  - rpcsts.idl
  - rpcbase.idl
  - rpcpvt.idl
  - rpcsts.idl
  - rpcbase.idl
  - rpcpvt.idl
  - rpcsts.idl
  - rpcbase.idl
  - rpcpvt.idl
  - rpcsts.idl
  - rpcbase.idl
  - rpcpvt.idl
  - rpcsts.idl

  In a PDS
  - IOVECTOR
  - LBASE
  - NBASE

- The input IDL and ACF files used by the z/OS DCE IDL compiler must be coded in code page IBM-1047. See the [z/OS DCE Administration Guide](https://www.ibm.com/support/knowledgecenter/SSQCHW_1.2.1.0/COM.CDF.1.2.1.0/c/com_zos_cdf_admin_guide_121.pdf) for more information.

- The z/OS DCE IDL compiler always produces output files (header files, client and server stub .c files) in code page IBM-1047.
Related Information

Books:

- z/OS DCE Application Development Guide: Core Components
- z/OS DCE Application Development Guide: Introduction and Style

Calling the IDL Compiler from TSO/E and Batch:  The following procedure is similar when calling the IDL compiler from batch.

When called from TSO/E and batch, the IDL compiler can access only PDS data sets.

The IDL compiler searches through the DD:USERACF data set for any related ACF. For example, if you compile a file named DD:USERIDL(SOURCE), the compiler searches for a file named DD:USERACF(SOURCE). The compiler also searches for any imported IDL file (and its related ACF).

The compiler searches for these files in the following order:

1. The DD:USERIDL data set. The compiler always searches this data set unless you specify the -no_def_idir and -I DD:ddname options together.
2. Any imported data set. The compiler searches each data set you are specifying in the -I DD:ddname option.
3. The system IDL data set DD:SYSSIDL. The compiler imports DD:SYSSIDL(NBASE), which resides in the system IDL data set. The compiler always searches this data set unless you specify the -no_def_idir option.
4. You can specify a prefix for the IDL data set using the -userpfx option (available in TSO/E only) as shown in the following example. If this option is unspecified, the default will be your user ID.
   
   \[
   \text{idl \ TEST \ -no\_cpp \ -keep \ c\_source \ -userpfx \ <\text{USERPRFX}>.\text{TEST}}
   \]

Calling the IDL Compiler from the OMVS Shell:  When called from the OMVS shell, the IDL compiler supports both HFS and PDS. The output file is stored in either an HFS file or a PDS.

To specify a PDS file or data set, prefix the PDS file or data set with // characters. The following example compiles test.idl from the current directory and saves the header in DCE.MYPROG.H(TEST)

\[
\text{idl \ test.idl \ -keep \ c\_source \ -I"//DCE.SYSIDL.IDL" \ -header"//DCE.MYPROG.H(\text{TEST})"}
\]

The IDL compiler searches through directories for any related attribute configuration file (ACF). For example, if you compile a file called source.idl, the compiler searches for a file named source.acf and also for any imported IDL file (and its related ACF). The compiler searches for these files in the following order:

1. The current working directory. The compiler always searches this directory.
2. Any imported directory. The compiler searches each directory you specify in the -Idirectory option.
   The IDL compiler does not search for an ACF file in a PDS whose name does not end in .ACF.
3. The system IDL directory. The compiler imports nbase.idl, which resides in the system IDL directory.
   The compiler always searches this data set unless you specify the -no_def_idir option.
4. The directory specified in the source file name. If you specify a directory in the source IDL path name, then that directory is searched for the corresponding ACF. In the following example, the IDL compiler searches for /path/pathname/my_source.acf, if my_source.acf is not found in any of the other directories.

\[
\$ \text{idl} \ /path/pathname/my\_source.idl}
\]
**Note:** For any imported IDL file and its corresponding ACF, the IDL compiler does not search the directory `/path/pathname`.

**Files:**
- `c89` UNIX System Services front end to IBM C/C++ for z/OS
- `<dceshared>/bin/idl` IDL Compiler
- `<dceshared>/include` System IDL directory for imported files
- `<dceshared>/include/dce/nbase.idl` Predefined IDL types
- `<dceshared>/nls/msg/LANG/dceidl.cat` Compiler error messages
- `<dceshared>/share/include/file.ext` All `.idl` or `.h` files that are part of z/OS DCE.

**Examples**

1. The following OMVS shell example generates stubs for `test.idl` from the current directory (`export` is pointing to the directory of `dce/export/mvs/usr/include/dce`). The C preprocessor is not run on the `.idl` file or any imported `.idl` file.
   ```bash
   idl -no_cpp -keep c_source -I export test.idl
   ```

2. The following OMVS shell example generates stubs for `greet.idl`. The IDL generated C source files are saved.
   ```bash
   idl greet.idl -keep c_source -cc_opt "-DMVS -D_DCE_THREADS -D_OPEN_SYS _W0,DLL"
   ```

3. The following OMVS shell example starts the IDL compiler to compile the interface definition file `test.idl` and keep the generated C source modules. Only server files are generated. The server stub default file name is overridden by creating a file named `test_ss.c` for the server stub module. The Manager Entry Point Vector is not defined in the generated server stub module.
   ```bash
   $ idl test.idl -keep c_source -client none -sstub test_ss.c -no_mepv
   ```

4. The following example compiles `test.idl` from the current directory and saves the header in `DCE.MYPROG.H(TEST)`.
   ```bash
   idl test.idl -keep c_source -I''/DCE.SYSIDL.IDL'' -header''/DCE.MYPROG.H(TEST)''
   ```

5. The following OMVS shell example compiles `test.idl` from the current directory and saves the output in `DCE.MYPROG.STUB`.
   ```bash
   idl test.idl -I''/DCE.SYSIDL.IDL'' -cstub ''/DCE.MYPROG.STUB(TESTCS)''
             -sstub ''/DCE.MYPROG.STUB(TESTSS)''
   ```

6. The following is a TSO/E example:
   ```bash
   idl TEST -no_cpp -keep c_source -userpfx <USERPRFX>.TEST
   ```

7. Use the JCL in this example to compile the IDL file from Batch mode. For more information on running the IDL compiler from Batch mode, see the [z/OS DCE Application Development Guide: Introduction and Style](#).
   ```bash
   //JOBNAME JOB (ACCOUNT)...your_job_parameters
   //**************************
   // RUN THE IDL COMPILER
   //**************************
   //IDLCOMP EXEC IDL,USERPFX='USERPRFX,<APPLNM>',
   //PARM='GREET -keep c_source'
   ```
import

Imports binding information and an object UUID from a server entry.

Format

```
rpccp import  starting-entry-name
  [-i if-id]
  [-e]
  [-o object-uuid]
  [-s syntax] [-u]
  [-n [integer]]
```

Options

- **-i** Defines an interface identifier to be imported (required). You can import only one interface at a time.
  
  The interface identifier has the following form:
  
  `interface-uuid,major-version,minor-version`
  
  The UUID is a hexadecimal string and the version numbers are decimal strings, for example:
  
  `-i ec1eeb60-5943-11c9-a309-08002b102989,1.1`
  
  Leading zeros in version numbers are ignored.

- **-e** Shows the name of the entry where the binding is found (optional).

- **-o** Declares the UUID of an object to be imported (optional). Only one UUID can occur in a single operation.
  
  If an object is specified, the import operation limits its search to server entries that contain both the specified interface identifier and object UUID when searching for a potential binding. Without the `-o` option, the import operation ignores object UUIDs.
  
  The UUID is a hexadecimal string, for example:
  
  `-o 3c6b8f60-5945-11c9-a236-08002b102989`

- **-s** Indicates the name syntax of the entry name (optional). The only value for this option is the `dce` name syntax, which is the default name syntax. Specifying the `-s` option is unnecessary.

- **-u** Updates the local CDS cache copy of name service data (optional).
  
  Name service data is cached locally on each machine in a cell. If a local cached copy of name service data satisfies a `rpccp` command, the local copy is used. However, a recent CDS update may not be included in locally cached copies of name service data. If the required data is not available in the local CDS cache, `rpccp` retrieves the required data from a CDS server.
  
  Specify the `-u` option to bypass the local cache, and go directly to a CDS server for the inquiry.

- **-n** Declares that the import operation is to continue until no more potential bindings are found (optional). Providing a numeric value to this option restricts the number of imported bindings. If you omit the `-n` option, only one binding is imported. If repeated, this operation may return the same binding.
import

For example, -n imports all available bindings, and -n 5 imports up to five bindings. The imported bindings are displayed as string bindings. If you omit this option, only one binding is imported.

If omitting the integer when specifying the -n option, ensure that it is the last argument specified, and that the starting-entry-name always precedes the -n.

Parameters

starting-entry-name
Indicates the name of the server entry where the import operation starts. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The import command imports binding information and an RPC object UUID for a specific RPC interface from a server entry. The name of the entry and the interface identifier are required. The entry name can refer to a server entry, a group, or a profile.

Privilege Required: You must have read permission to the specified CDS object entry (the starting name service entry) and to any CDS object entry in the resulting search path.

Note: The import command is supported in z/OS DCE by the dcecp rpcentry import command.

Examples

The following commands run RPCCP and import an interface and object:

READY
rpccp
rpccp> import ./LandS/anthro/Cal_host_3 \\
> -i ec1eeb6wzerodot-5943-11c9-a309-08002b102989,1.1 \\
> -o 3dbeeawzerodot-fb6c-11c9-8eea-08002b0F4528 \\
> -n

Related Information

Commands:

export quit show server unexport
ping

Checks if the target server is listening.

Format

rpccp ping server-address [-w count]

Options

-w Indicates the number of times to ping the server. If no option is specified, the server is pinged once.

Parameters

server-address This is the string binding of the target server, for example:

ncadg_ip_udp:9.21.22.196[135]

The string binding must contain a protocol sequence, a network address for the remote system, and an endpoint.

count Indicates the number of times to ping the server.

Usage

The ping command checks whether the target server is listening or not.

Examples

The following command pings the server ncadg_ip_udp:9.21.22.196[135] twice:

Output Messages

The command will echo a message to inform the user that RPCCP is pinging a server:

- **Pinging server ncadg_ip_udp:9.21.22.196[135].**

Then will return a number of response messages corresponding to the requested count. Responses can be:

- **Ping server response took xxx msec.**
- **Ping server timed out after xxxx msec.**
query epdb

Queries the state of the endpoint map database file and returns the total number of active server entries registered.

Format

rpccp query epdb [ host-address ]

Parameters

host-address  This parameter is a string binding that indicates where to find the target endpoint map. When accessing the local endpoint map, you can specify what protocol sequence to use (optional), for example:

ncadg_ip_udp:
When accessing a remote endpoint map, you must specify both a protocol sequence and a network address for the remote system (required), for example:

ncadg_ip_udp:9.21.21.91
An endpoint is unnecessary in local or remote host addresses, and the query epdb command ignores any endpoint specified as part of a host address.

Usage

The query epdb command queries the state of the endpoint map database file and returns the total number of active server entries registered.

Privilege Required:  The calling principal must have l (list) or c (control) permission to the endpoint map database file to perform this operation.

Examples

The following example shows the use of the query epdb command specifying both a protocol sequence and a network address.

READY
rpccp query epdb ncadg_ip_udp:9.21.21.91

Output Messages

For successful operations, the command returns two messages. The first message is one of either:

- End point database file detached
- End point database file exists.

The second message is:

- Total number of active registered entries is xx.

Otherwise, a message is returned with an error code.
Related Information

Commands:

rebuild epdb
quit

quit

Causes the DCE RPC control program to end.

Format

rpccp quit

Usage

The quit command causes the rpccp program to end. The original user environment is restored.

Examples

The following example shows the use of the quit command if rpccp was entered at the system prompt:

READY
rpccp
rpccp> quit
READY

Related Information

Commands:

exit
**rebuild epdb**

Rebuilds the endpoint map database file from the DCED host daemon end point map in-memory entries.

**Format**

`rpccp rebuild epdb [ host-address ]`

**Parameters**

*host-address*  
This parameter is a string binding that indicates where to find the target endpoint map. When accessing the local endpoint map, you can specify what protocol sequence to use (optional), for example:

```
ncadg_ip_udp:
```

When accessing a remote endpoint map, you must specify both a protocol sequence and a network address for the remote system (required), for example:

```
ncadg_ip_udp:9.21.21.91
```

An endpoint is unnecessary in local or remote host addresses, and the `rebuild epdb` command ignores any endpoint specified as part of a host address.

**Usage**

The `rebuild epdb` command rebuilds the endpoint map database file from the DCE host daemon end point map in-memory entries.

**Privilege Required:**  
A DCE principal must have *x* (execute) or *c* (control) permission to the endpoint map database file to perform this operation.

**Examples**

The following example shows the use of the `rebuild epdb` command specifying both a protocol sequence and a network address.

```
READY
rpccp rebuild epdb ncadg_ip_udp:9.21.21.91
```

**Output Messages**

The command returns one of the following messages, depending on whether the operation is successful:

- **Endpoint database rebuilt successfully.**
- **RPC control program cannot rebuild endpoint map database.**
  
  DCE status code: 0x1xxxxxxx

**Related Information**

Commands:

`query epdb`
rpccp

Purpose

Starts the RPC control program.

Format

rpccp [rpccp-command]

Arguments

rpccp-command Specifies one of the following control program commands:

add element Adds an element to a profile in a name service entry; if the specified entry does not exist, creates the entry.
add entry Adds an entry to the name service database.
add member Adds a member to a group in a name service entry; if the specified entry does not exist, creates the entry.
exit Leaves the RPC control program.
export Exports binding information for an interface identifier or object UUIDs or both to a server entry; if the specified entry does not exist, creates the entry.
help Displays a list of commands or the possible options of a specified command.
import Imports binding information and an object UUID from a server entry.
quit Leaves the RPC control program.
remove element Removes selected elements from a profile.
remove entry Removes an entry from the name service database.
remove group Removes all group members and the group from the specified entry.
remove mapping Removes specified elements from the local endpoint map or from the endpoint map of a specified remote host.
remove member Removes a selected member from a group.
remove profile Removes all profile elements and the profile from the specified entry.
show entry Shows the NSI attributes of an entry.
show group Shows the members of a group.
show mapping Shows the elements of the local endpoint map.
show profile Shows the elements of a profile.
show server Shows the binding information, interface identifier, and object UUIDs in a server entry.
unexport Removes binding information, interface identifiers, and object UUIDs from a server entry.
Note: A server entry equates to an NSI binding attribute and, optionally, an object attribute; a group equates to an NSI group attribute; and a profile equates to an NSI profile attribute. Typically, each server’s entries, groups, and profiles reside in distinct name service entries.

Usage

Provides a set of commands for managing name service for RPC applications and for managing the endpoint map.

This facility is superseded by the DCE control program (dcecp) for z/OS DCE. With the exception of the following subcommands, the rpccp function is supported in z/OS DCE by the dcecp command.

- rpccp help
- rpccp ping
- rpccp query epdb
- rpccp rebuild epdb

You can start the control program command in three ways:

- From inside the control program:

  You can start and enter the control program using the rpccp command alone, without any argument. The control program then displays the control program prompt (rpccp>):

  ```
  READY
  rpccp
  rpccp>
  ```

  You can then enter any control program command, for example:

  `rpccp> show entry/.:/LandS/anthro/pr_server_node3`

  **Note:** If the DCE servers are not running, the rpccp commands cannot run successfully. If this is the case and you start rpccp, it will still display the rpccp> prompt. You will receive an error message when you subsequently run the rpccp subcommands.

  You can leave the control program and return to the system prompt using the exit or quit command.

  If you enter input that is not valid, the control program displays the valid commands.

- From the system prompt:

  Interactively or in a command procedure, enter the rpccp command with an internal command of the control program as the first argument. For example, you can enter the show entry command:

  ```
  READY
  rpccp show entry/.:/LandS/anthro/pr_server_node3
  ```

  **Note:** The maximum number of characters allowed in the argument is 100.

- From JCL:

  You can enter the internal control program commands either as program parameters, or as SYSSIN data hard coded in the JCL, or as text entries in a user-defined file redirected to the SYSSIN stream. The last two methods to enter the internal control program commands are most effective when you have to enter a large number of commands or data entries. See "JCL Examples" on page 478 for different ways to start internal program commands with JCL.

  Enter all non-interactive rpccp commands from native TSO/E or the shell only. If these commands and their arguments are entered from the ISPF command line they are converted to lowercase characters. This is a problem for arguments that have mixed or all uppercase characters. For example, the following
command actually adds the entry ./lands/anthro/cal_host_2 and not the entry
./LandS/anthro/Cal_host_2:

tso rpccp add entry ./LandS/anthro/Cal_host_2

If you run rpccp in batch mode, the input to the control programs and the output for each command are
written to the job execution log. You can then determine which input commands were successful and
which were not.

Options: Except for the exit and quit commands, rpccp commands have one or more options. Each
option is identified by a dash (-) followed by a letter; for example, -s. Some options require arguments.

Commands that access NSI operations also require the name of a name service entry as an argument.
The order of arguments and the entry-name option are arbitrary; for example, the following placements of
arguments and options are equivalent:

```
rpccp> add element ./LandS/anthro/mis_node_2 \ 
> -i ec1eeb6wzerodot-5943-11c9-a309-08002b102989,1.0

rpccp> add element -i ec1eeb60-5943-11c9-a309-08002b102989,1.0 \ 
> ./LandS/anthro/mis_node_2
```

Continuation Character: The back slash character (\) can be used at the end of a line that exceeds 255 characters, to prompt for continuation of the internal command line. Use the backslash character \ as the last non-blank character on the line to be continued, and continue the command on the
next line.

**RPC Control Program Commands**

The RPC control program commands are:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add element</td>
<td>Adds an element to a profile in a name service entry</td>
</tr>
<tr>
<td>add entry</td>
<td>Adds a name service entry to the name service database</td>
</tr>
<tr>
<td>add mapping</td>
<td>Adds or replaces server address information in the local endpoint map</td>
</tr>
<tr>
<td>add member</td>
<td>Adds a member to a group in a name service entry</td>
</tr>
<tr>
<td>exit</td>
<td>Ends the RPC control program</td>
</tr>
<tr>
<td>export</td>
<td>Exports binding information for an interface identifier, object UUIDs, or both to a server entry</td>
</tr>
<tr>
<td>help</td>
<td>Starts the rpccp help service</td>
</tr>
<tr>
<td>import</td>
<td>Imports binding information and an object UUID from a server entry</td>
</tr>
<tr>
<td>ping</td>
<td>Checks if the target server is listening</td>
</tr>
<tr>
<td>query epdb</td>
<td>Queries the state of the endpoint map database file and returns the total number of server entries</td>
</tr>
<tr>
<td>quit</td>
<td>Ends the DCE RPC control program session</td>
</tr>
<tr>
<td>rebuild epdb</td>
<td>Rebuilds the endpoint map database file from the RPCD in-memory entries</td>
</tr>
<tr>
<td>remove element</td>
<td>Removes selected elements from a profile</td>
</tr>
<tr>
<td>remove entry</td>
<td>Removes a name service entry from the name service database</td>
</tr>
<tr>
<td>remove group</td>
<td>Removes all group members and the group from the specified name service entry</td>
</tr>
<tr>
<td>remove mapping</td>
<td>Removes specified elements from the local endpoint map</td>
</tr>
</tbody>
</table>
Table 9 (Page 2 of 2). RPCCP Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>remove member</td>
<td>Removes a specified member from a group</td>
</tr>
<tr>
<td>remove profile</td>
<td>Removes all profile elements and the profile from the specified name service entry</td>
</tr>
<tr>
<td>show entry</td>
<td>Shows the NSI attributes of a name service entry</td>
</tr>
<tr>
<td>show group</td>
<td>Shows the members of a group</td>
</tr>
<tr>
<td>show mapping</td>
<td>Shows the elements of either a local endpoint map or a remote endpoint map</td>
</tr>
<tr>
<td>show profile</td>
<td>Shows the elements of a profile</td>
</tr>
<tr>
<td>show server</td>
<td>Shows the binding information, interface identifiers, and object UUIDs in a server entry</td>
</tr>
<tr>
<td>unexport</td>
<td>Removes binding information, interface identifiers, and object UUIDs from a server entry</td>
</tr>
<tr>
<td>uuidgen</td>
<td>Generates a Universal Unique Identifier</td>
</tr>
</tbody>
</table>

For more information on any of the above **rpccp** commands, see the appropriate page in this chapter.

The following table shows the **dcecp** commands that replace the **rpccp** commands:

Table 10 (Page 1 of 2). DCECP Equivalent Commands for RPCCP Commands

<table>
<thead>
<tr>
<th>rpccp Command</th>
<th>dcecp Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpccp</td>
<td>dcecp</td>
</tr>
<tr>
<td>add element</td>
<td>rpcprofile add</td>
</tr>
<tr>
<td>add entry</td>
<td>rpcentry create</td>
</tr>
<tr>
<td>add mapping</td>
<td>endpoint create</td>
</tr>
<tr>
<td>add member</td>
<td>rpcgroup add</td>
</tr>
<tr>
<td>exit</td>
<td>exit</td>
</tr>
<tr>
<td>export</td>
<td>rpcentry export</td>
</tr>
<tr>
<td>help</td>
<td>dcecp help</td>
</tr>
<tr>
<td>import</td>
<td>rpcentry import</td>
</tr>
<tr>
<td>ping</td>
<td>None</td>
</tr>
<tr>
<td>query epdb</td>
<td>None</td>
</tr>
<tr>
<td>quit</td>
<td>quit</td>
</tr>
<tr>
<td>rebuild epdb</td>
<td>None</td>
</tr>
<tr>
<td>remove element</td>
<td>rpcprofile remove</td>
</tr>
<tr>
<td>remove entry</td>
<td>rpcentry delete</td>
</tr>
<tr>
<td>remove group</td>
<td>rpcgroup delete</td>
</tr>
<tr>
<td>remove mapping</td>
<td>endpoint delete</td>
</tr>
<tr>
<td>remove member</td>
<td>rpcgroup remove</td>
</tr>
<tr>
<td>remove profile</td>
<td>rpcprofile delete</td>
</tr>
<tr>
<td>show entry</td>
<td>rpcentry show</td>
</tr>
<tr>
<td>show group</td>
<td>rpcgroup show</td>
</tr>
<tr>
<td>show mapping</td>
<td>endpoint show</td>
</tr>
<tr>
<td>show profile</td>
<td>rpcprofile show</td>
</tr>
<tr>
<td>show server</td>
<td>rpcentry show</td>
</tr>
</tbody>
</table>
Scope of the RPC Control Program Commands: The following table describes the scope of the RPC control program commands.

<table>
<thead>
<tr>
<th>Scope</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Entries</td>
<td>add entry</td>
</tr>
<tr>
<td></td>
<td>remove entry</td>
</tr>
<tr>
<td></td>
<td>show entry</td>
</tr>
<tr>
<td>Server entry</td>
<td>export</td>
</tr>
<tr>
<td></td>
<td>import</td>
</tr>
<tr>
<td></td>
<td>show server</td>
</tr>
<tr>
<td></td>
<td>unexport</td>
</tr>
<tr>
<td>Group</td>
<td>add member</td>
</tr>
<tr>
<td></td>
<td>remove group</td>
</tr>
<tr>
<td></td>
<td>remove member</td>
</tr>
<tr>
<td></td>
<td>show group</td>
</tr>
<tr>
<td>Profile</td>
<td>add element</td>
</tr>
<tr>
<td></td>
<td>remove element</td>
</tr>
<tr>
<td></td>
<td>remove profile</td>
</tr>
<tr>
<td></td>
<td>show profile</td>
</tr>
<tr>
<td>Endpoint map</td>
<td>add mapping</td>
</tr>
<tr>
<td></td>
<td>remove mapping</td>
</tr>
<tr>
<td></td>
<td>show mapping</td>
</tr>
<tr>
<td>Endpoint map database</td>
<td>rebuild epdb</td>
</tr>
<tr>
<td></td>
<td>query epdb</td>
</tr>
</tbody>
</table>
Environment Variables: When RPCCP is started, you can set the following environment variables:

DCE RPC environment variables:

**RPC_DEFAULT_ENTRY_SYNTAX**

The DCE name syntax is the only syntax supported by the DCE Cell Directory Service (CDS). However, the Name Service Interface (NSI) is independent of any specific name service. You can override this default in any NSI command of the control program by using the `-s` option to specify an alternative entry syntax.

To call the `RPC_DEFAULT_ENTRY_SYNTAX`, you must specify the integer 3 to indicate the DCE syntax.

To set `RPC_DEFAULT_ENTRY_SYNTAX`, use the `name=value` command to define an environment variable. The following example specifies `dce` as the default name syntax in a login command file:

```bash
# login command file
# setting dce as default name syntax
RPC_DEFAULT_ENTRY_SYNTAX=3
```

**RPC_DEFAULT_ENTRY**

For the `import` command, you can use this environment variable to indicate the entry where the search operation starts. Usually, the starting entry is a profile. Refer to the [z/OS DCE Administration Guide](https://www.ibm.com) for z/OS considerations.

The Name Service Interface: The DCE RPC NSI is independent of any particular name service. CDS, however, is the only name service available. For more details on the name service interface, see the NSI usage chapter in the [z/OS DCE Application Development Guide: Core Components](https://www.ibm.com). For a description of the DCE Cell Directory Service, see the [z/OS DCE Administration Guide](https://www.ibm.com).

Name Service Entries: To store information about RPC servers, interfaces, and objects, the NSI defines several name service entries:

- A server entry stores binding information, interface identifiers, and object UUIDs for an RPC server.
- A group corresponds to one or more RPC servers that offer a common RPC interface, type of RPC object, or both.
- A profile defines search paths for looking in a name service database for a server that offers a particular RPC interface and object.

Note: When the NSI is used with the Cell Directory Service, the name service entries are CDS object entries.

Structure of Entry Names: Each entry in a name service database is identified by a unique global name made up of a cell name and a cell-relative name.

A cell is a group of users, systems, and resources that share common DCE services. A cell configuration includes at least one cell directory server, one security server, and one time server. A cell's size can range from one system to thousands of systems. For information on cells, see the CDS section of the [z/OS DCE Administration Guide](https://www.ibm.com).

The following is an example of a global name:

```
/.../C=US/O=uw/OU=MadCity/LandS/anthro/Stats_host_2
```

The parts of a global name are as follows:

- Cell name (using X.500 name syntax), for example:
rpccp

/.../C=US/O=uw/OU=MadCity

The symbol /... begins a cell name. The letters before the equal signs (=) are abbreviations for country (C), organization (O), and organization unit (OU).

For entries in the local cell, you can represent the cell name by a ./: prefix, in place of the actual cell name, for example,

/./:LandS/anthro/Stats_host_2

For NSI operations on entries in the local cell, you can omit the cell name.

- Cell-relative name

Each name service entry requires a cell-relative name, which contains a directory path name and a leaf name.

A directory path name follows the cell name and indicates the hierarchical relationship of the entry to the cell root.

The directory path name is the middle portion of the global name. The cell name is to the left of the directory path name, and the leaf name is to the right:

cell-name + directory-path-name + leaf-name

The directory path name contains the names of any subdirectories in the path; each subdirectory name begins with a slash (/):

/sub-dir-a-name/sub-dir-b-name/sub-dir-c-name

Directory paths are created by name service administrators. If an appropriate directory path does not exist, ask your name service administrator to extend an existing path or create a new path. In a directory path, the name of a subdirectory should reflect its relationship to its parent directory (the directory that contains the subdirectory).

A leaf name identifies the specific entry.

The leaf name is the right-hand part of the global name beginning with the rightmost slash.

In the following example, /.../C=US/O=uw/OU=MadCity is the cell name, /LandS/anthro is the directory path name, and /Cal_host_4 is the leaf name:

/.../C=US/O=uw/OU=MadCity/LandS/anthro/Cal_host_4

If a name service entry is located at the cell root, the leaf name directly follows the cell name, for example, ./:/cell-profile.

Note: When the NSI is used with CDS, the cell-relative name is a CDS name.

Guidelines for Constructing Names of Name Service Entries: A global name includes both a cell name and a cell-relative name composed of a directory path name and a leaf name. The cell name is assigned to a cell root at its creation. When you specify only a cell-relative name to an NSI command, the NSI automatically expands the name into a global name by inserting the local cell name. When returning the name of a name service entry, a group member, or a member in a profile element, NSI operations return global names.

The directory path name and leaf name uniquely identify a name service entry. The leaf name should somehow describe the entry, for example, by identifying its owner or its contents. The remainder of this section contains guidelines for choosing leaf names.

Note: Directory path names and leaf names are case sensitive.
Naming a Server Entry

For a server entry that advertises an RPC interface or service offered by a server, the leaf name must distinguish the entry from the equivalent entries of other servers. When a single server instance runs on a host, you can ensure a unique name by combining the name of the service, the interface (from the interface definition), or the system name for the server's host system.

For example, consider two servers, one offering a calendar service on host JULES, and one on host VERNE.

The server on JULES uses the following leaf name:

calendar_JULES

The server on VERNE uses the following leaf name:

calendar_VERNE

For servers that perform tasks on or for a specific system, an alternative approach is to create server entries in a system-specific host directory within the name service database. Each host directory takes the name of the host to which it corresponds. Because the directory name identifies the system, the leaf name of the server entry name need not include the host name, for example:

/./LandS/host_1/Process_control

To construct names for the server entries used by distinctive server instances on a single host, you can construct unique server entry names by combining the name of the server's service, interface, or object; the system name of the server's host system; and a reusable instance identifier, such as an integer.

For example, the following leaf names distinguish two instances of a calendar service on the JULES system:

calendar_JULES_01

calendar_JULES_02

Avoid automatically generating entry names for the server entries of server instances, for example, by using unique data such as a time stamp (calendar_verne_15OCT91_21:25:32) or a process identifier (calendar_jules_208004D6). When a server incorporates such unique data into its server entry names, each server instance creates a separate server entry, causing many server entries. When a server instance stops running, it leaves an obsolete server entry that is not reused. The creation of a new entry whenever a server instance starts may impair performance.

A server can use multiple server entries to advertise different combinations of interfaces and objects. For example, a server can create a separate server entry for a specific object (and the associated interfaces). The name of such a server entry should correspond to a well-known name for the object. For example, consider a server that offers a horticulture bulletin board known to users as horticulture_bb. The server exports the horticulture_bb object, binding information, and the associated bulletin-board interface to a server entry whose leaf name identifies the object, as follows:

horticulture_bb

Note: An RPC server that uses RPC authentication can choose identical names for its principal name and its server entry. Use of identical names permits a client that calls the rpc_binding_set_auth_info routine to automatically determine a server's principal name. The client will assume the principal name to be the same as the server's entry name. If a server uses different principal and server entry names, users must explicitly supply the principal name. For an explanation of principal names, see the DCE
Naming a Group

The leaf name of a group should indicate the interface, service, or object that determines membership in the group. For example, for a group whose members are selected because they advertise an interface named Statistics, the following is an effective leaf name:

Statistics

For a group whose members advertise laser-printer print queues as objects, the following is an effective leaf name:

laser-printer

Naming a Profile

The leaf name of a profile should indicate the profile users. For example, for a profile that serves the members of an accounting department, the following is an effective leaf name:

accounting_profile

Privilege Required: To use the NSI commands to access entries in a CDS database, you must have access control list (ACL) permissions. Depending on the NSI operation, you must have ACL permissions to the parent directory, or the CDS object entry (the name service entry), or both. The ACL permissions are:

- To create an entry, you must have insert permission to the parent directory.
- To read an entry, you must have read permission to the CDS object entry.
- To write to an entry, you must have write permission to the CDS object entry.
- To delete an entry, you must have delete permission either to the CDS object entry or to the parent directory.

Note: Write permission does not imply read permission.

The required ACL permissions for each control program NSI command are discussed in the Privilege Required section of each command.

JCL Examples: The following examples show the different ways to start internal control program commands with JCL:
Execute the RPC control program using JCL. This example shows the use of parameter as input to RPCCP. A DCE login using JCL must be completed before running RPCCP. Note: You must check output for error messages if any.

```
//RPCCP EXEC PROC=RPCCP,
// PARM='show mapping'
// Note: The data set below must be allocated by the user and is the same one used when the DCE Login JCL is submitted.
//RPCCP.DCEENV DD DSN='user1.SECCCENV.RPCSERV',DISP=SHR
```

Chapter 3. Remote Procedure Call
rpccp

//user1 JOB (9998,B0), 'RPCCP RUN', CLASS=A, MSGCLASS=S, NOTIFY=user1, USER=user1, PASSWORD=?????????, MSGLEVEL=(1,1)
//JOPARMS LINES=99, TIME=2

//** Execute the RPC control program using JCL. *
//** This example shows the use of file data as SYSIN data stream. *
//** A DCE login using JCL must be completed before running RPCCP. *
//** Note: You must check output for error messages if any. *

//** EXEC PROC=RPCCP

//** Note: The data set below must be allocated by the user and is the same one used when the DCE Login JCL is submitted. *

//** RPCC.PCEENV DD DSN='user1.SECCCENV.RPCSERV',DISP=SHR

//** RPCCP internal commands can be entered as a list of commands *
//** in the user defined file. *
//** Sample content of file: *
//** add entry /.:/my_entry *
//** export -i EC1EEB60-5943-11C9-A3Rzerodot9-Rzerodot8Rzerodot2B1Rzerodot2989,1.1 \
//** -b ncadg_ip_udp:9.21.2Rzerodot.96 \
//** /.:/my_entry *
//** show entry /.:/my_entry *
//** remove entry /.:/my_entry *
//** exit *

//** SYSIN DD DSN=user1.RPCCPCMD,DISP=SHR

Related Information: RPCCP Commands:
add element help remove element show entry
add entry import remove entry show group
add mapping ping remove group show mapping
add member query epdb remove mapping show profile
rebuild epdb
exit
export

480 Command Reference
remove element

Removes selected elements from a profile.

Format

rpccp remove element profile-entry-name
(-d | -i if-id -m member | -a annotation)
[-s syntax]

Options

- **d**
  Removes the default profile element. With the -d option, the -a, -i, and -m options are ignored.

- **i**
  Defines an interface identifier for the profile element to be removed for a member specified with the -m option. Only one interface and member pair can be removed in a single operation. If you supply multiple instances of the -i option, the command uses the final instance.

  The -i and -m options take precedence over the -a option. However, if the default profile element is specified (by the -d option), the -i and -m options are ignored.

  The interface identifier value has the following form:

  \[interface-uuid[major-version.minor-version]\]

  \[major-version.minor-version\] is optional; defaults to 0,0.

  The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:

  -i ec1eeb60-5943-11c9-a390-0802b102989,1.1

  Leading zeros in version numbers are ignored.

- **m**
  Defines a member name for the profile element to be removed. This option is required if the interface identifier is specified. Only one interface and member can be removed in a single operation. If you supply multiple instances of the -m option, the command uses the final instance.

- **a**
  Removes all elements whose annotation fields match the specified annotation; in the presence of the -d option or -i and -m options, the -a option is ignored.

  Note that the shell supports quotation marks around the annotation field of profile elements to allow you to include internal spaces in an annotation. The control program does not support quotation marks around the annotation field. To specify or refer to annotations from within the control program, limit each annotation to an unbroken alphanumeric string, for example, CalendarGroup. To refer to annotations from the system prompt, do not incorporate quotation marks into any annotation.

- **s**
  Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

Parameters

profile-entry-name

Indicates the name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.
remove element

Usage

The **remove element** command removes an element from a profile in the name service database. For a description of the fields in a profile element, see the description for the **RPCCP add element** command.

The **remove element** command requires the entry name of the profile and one of the following options or option combinations: `-d`, `-i` and `-m`, or `-a`.

**Privilege Required:** You must have read permission and write permission to the CDS object entry (the target profile entry).

**Note:** The **remove element** command is supported in z/OS DCE by the **dcep rpcprofile remove** command.

Examples

Run RPCCP, and remove an element from a profile:

```
READY
rpccp
rpccp> remove element -i ec1eeb60-5943-11c9-a309-08002b102989,1.1
    > -m ./LandS/anthro/Calendar_group \
    > ./LandS/anthro/molly_o_profile
```

Related Information

Commands:

```text
add element remove profile show profile
```
remove entry

Removes a name service entry from the name service database.

Format

rpccp remove entry  entry-name
[-s syntax]

Options

-s Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

Parameters

entry-name Indicates the name of the target name service entry. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The remove entry command removes an entry from the name service database. The name of the entry is required.

Privilege Required: You must have read permission to the CDS object entry (the target name service entry). You also require delete permission to the CDS object entry or to the parent directory.

Note: The remove entry command is supported in z/OS DCE by the dcecp rpcentry delete command.

Examples

The following commands run RPCCP and remove the entry ./LandS/anthro/Cal_host_2 from the local cell of the name service database:

READY
rpccp
rpccp> remove entry
././LandS/anthro/Cal_host_2

Related Information

Commands:

add entry        show entry
remove group

Removes all group members and the group from the specified name service entry.

Format

rpccp remove group  group-entry-name
[-s syntax]

Options

-s  Indicates the name syntax of the entry name (optional). The only value for this option is
the dce name syntax, which is the default name syntax. Specifying the -s option is
unnecessary.

Parameters

group-entry-name  Indicates the name of the target group. For an entry in the local cell, you can omit the
cell name and specify only the cell-relative name.

Usage

The remove group command removes a group from the name service database. The group need not be
empty. The entry name of the group is required.

Privilege Required: You must have write permission to the CDS object entry (the target group
entry).

Note: The remove group command is supported in z/OS DCE by the dcecp rpcgroup delete
command.

Examples

The following commands run RPCCP and remove the group from the name service entry
/./LandS/anthro/Calendar_group:

READY
rpccp
rpccp> remove group
/./LandS/anthro/Calendar_group

Related Information

Commands:

add member  remove member  show group
remove mapping

Removes specified elements from the local endpoint map.

Format

```
rpccp remove mapping
   -b string-binding [-b...] -i if-id
   [-o object-uuid [-o...]]
```

Options

- **-b**
  Declares a string binding (required) over which the server can receive remote procedure calls. You must also specify an interface identifier (using the `-i` option). Each command accepts up to 32 `-b` options.
  
The value has the form of an RPC string binding, without an object UUID, for example:
  
  ```
  -b ncadg_ip_udp:63.0.2.17[5347]
  ```
  
  Note that depending on your system, string binding delimiters such as brackets (`[]`) may need to be preceded by an escape symbol (`\`) or placed within quotation marks (`'` or `"`) at the system prompt. Requirements vary from system to system, and you must conform to the usage rules of your system.

- **-i**
  Declares an interface identifier (required) to remove from the local endpoint map. Only one interface can be removed in a single operation. The interface identifier has the following form:
  
  ```
  interface-uuid[major-version.minor-version]
  ```
  
  `major-version.minor-version` is optional; defaults to 0,0.
  
The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:
  
  ```
  -i ec1eeb6wzerodot-5943-11c9-a3wzerodot9-wzerodot8wzerodotwzerodot2b1wzerodot2989,1.1
  ```
  
  Leading zeros in version numbers are ignored.

- **-o**
  Defines an object UUID that further determines the endpoint map elements that are removed (optional). Each `remove mapping` command accepts up to 32 `-o` options.
  
The UUID is a hexadecimal string, for example:
  
  ```
  -o 3c6b8f60-5945-11c9-a236-08002b102989
  ```

Usage

The `remove mapping` command removes server address information from the local endpoint map. Each map element corresponds to an object UUID, interface identifier, annotation (optional), and binding information.

This command requires one interface identifier (the `-i` option), at least one string binding (the `-b` option), and optionally, one or more object UUIDs (the `-o` option). Each instance of the command accepts from 1 to 32 `-b` options and from 0 to 32 `-o` options. The options work together to delimit the elements to be removed from the target endpoint map. The command removes any map element that contains the specified interface identifier, a specified string binding, and a specified object UUID (if any).
remove mapping

Note: The **remove mapping** command is supported in z/OS DCE by the **dcecp endpoint delete** command.

**Examples**

The following command operates from the system prompt to remove a map element from the local endpoint map. The command removes only the map element that contains the specified interface identifier, server address (specified as a string binding), and object UUID.

```
READY
rpccp remove mapping
> -b ncadg_ip_udp:16.20.16.64[3424] \
> -i ec1eeb60-5943-11c9-a3b9-08002b102989,1.1 \n> -o 30db8888-1b6c-11c9-8ee8-08002b0f4528
READY
```

**Related Information**

Commands:

- **add mapping**
- **show mapping**
- **show server**
remove member

Removes a specified member from a group.

Format

rpccp remove member  group-entry-name
-m  member
[-s  syntax]

Options

-m  Declares the entry name of the group member to be removed (required).
-s  Indicates the name syntax of the entry name (optional). The only value for this option is
     the dce name syntax, which is the default name syntax. Specifying the -s option is
     unnecessary.

Parameters

group-entry-name  Indicates the name of the target group. For an entry in the local cell, you can omit the
cell name and specify only the cell-relative name.

Usage

The remove member command removes a specified member from a specified group.

Privilege Required:  You must have read permission and write permission to the CDS object entry
                   (the target group entry).

Note:  The remove member command is supported in z/OS DCE by the dcecp rpcgroup remove
command.

Examples

The following TSO commands run RPCCP and remove the member /./LandS/anthro/Cal_host_2 from the
group /./LandS/dept/Calendar_group:

rpccp
 rpccp> remove member  \    
>  -m  /./LandS/anthro/Cal_host_2  \    
>  /./LandS/anthro/Calendar_group

The following command removes the member /./LandS/anthro/Cal_host_3 from the group
/./LandS/anthro/Calendar_group:

READY
 rpccp remove member  \    
>  -m  /./LandS/anthro/Cal_host_3  \    
>  /./LandS/anthro/Calendar_group
remove member

Related Information

Commands:

add member  remove group  show group
remove profile

Removes all profile elements and the profile from the specified name service entry.

Format

rpccp remove profile profile-entry-name
[-s syntax]

Options

-s  Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

Parameters

profile-entry-name  Indicates the name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The remove profile command removes a profile (and all of its elements) from the name service database. The entry name of the profile is required.

Privilege Required:  You must have write permission to the CDS object entry (the target profile entry).

Note:  The remove profile command is supported in z/OS DCE by the dcecp rpcprofile delete command.

Examples

The following commands run RPCCP and remove the profile named ./LandS/anthro/molly_o_profile:

READY
rpccp
rpccp> remove profile \  
> ./LandS/anthro/molly_o_profile

Related Information

Commands:

add element  remove element  show profile
show entry

Shows the NSI attributes of a name service entry.

Format

rpccp show entry entry-name
[-i if-id] [-s syntax] [-u]

Options

- **i**
  Selects a specified interface identifier (optional). Only elements containing that identifier are shown. The interface identifier value has the following form:

  interface-uuid[major-version.minor-version]

  major-version.minor-version is optional; defaults to 0,0.

  The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:

  -i ec1eeb60-5943-11c9-a309-08002b102989,1.1

  Leading zeros in version numbers are ignored.

- **s**
  Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

- **u**
  Updates the local CDS cache copy of name service data (optional).

  Name service data is cached locally on each machine in a cell. If an rpccp inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the required data is not available in the local CDS cache, rpccp goes to a CDS server(s) to retrieve the required data. Then rpccp updates the local CDS cache.

  Using the -u option bypasses the local cache, allowing rpccp to go directly to a CDS server for the inquiry. Then rpccp updates the local CDS cache.

Parameters

entry-name

Indicates the name of the target name service entry. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The show entry command shows the NSI attributes of a name service entry. The name of the entry is required.

Note that this operation shows all of the compatible bindings for a given interface.

The show entry command shows the same list of string bindings as the import operation returns for the specified entry. This list includes all string bindings that refer to a major version that matches the specified version, and a minor version that is equal to or greater than the specified version. The list may include string bindings exported for other versions of the interface that are upwardly compatible, rather than for this particular version of the interface.
Privilege Required:  You must have read permission to the CDS object entry (the target name service entry).

Note:  The **show entry** command is supported in z/OS DCE by the **dcecp rpcentry show** command.

## Examples

The following command operates in TSO command mode to show service entry
`/./LandS/anthro/calendar_mgr_node_3`.

```
rpccp show entry /./LandS/anthro/Cal_host_3
```

The following commands run the control program and show the name service entry`/./LandS/anthro/Calendar_group`:

```
READY
rpccp
rpccp> show entry \
> /./LandS/anthro/Calendar_group
```

## Related Information

Commands:

- **add entry**
- **remove entry**
show group

Shows the members of a group.

Format

\texttt{rpccp show group \ group-entry-name}
\[\texttt{[\texttt{-m member}] \texttt{[\texttt{-r [integer]}}}\]
\[\texttt{[\texttt{-s syntax}] \texttt{[\texttt{-u}]}}\]

Options

\texttt{-m} \quad Declares the name of a single group member.

\texttt{-r} \quad Indicates that the \texttt{show group} operation is recursive. If any members of a group are also groups, their entries are shown. By default, the \texttt{-r} option causes the \texttt{show group} operation to be called until all nested groups are expanded. For example, \texttt{-r} shows the members of the specified group and all nested groups.

You can limit recursion to one or more levels by specifying a decimal integer as part of the \texttt{-r} option. For example, \texttt{-r 1} shows the members of the specified group and, for members that are groups, the command also shows their members; then recursion stops.

Without the \texttt{-r} option, only the members of the specified group are shown.

\texttt{-s} \quad Indicates the name syntax of the entry name (optional). The only value for this option is the \texttt{dce} name syntax, which is the default name syntax. Specifying the \texttt{-s} option is unnecessary.

\texttt{-u} \quad Updates the local CDS cache copy of name service data (optional). Name service data is cached locally on each machine in a cell. If an \texttt{rpccp} inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the required data is not available in the local CDS cache, \texttt{rpccp} goes to a CDS server(s) to retrieve the required data. Then \texttt{rpccp} updates the local CDS cache.

Using the \texttt{-u} option bypasses the local cache, allowing \texttt{rpccp} to go directly to a CDS server for the inquiry. Then \texttt{rpccp} updates the local CDS cache.

Parameters

\texttt{group-entry-name}

Indicates the name of the target group. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The \texttt{show group} command shows the members of a group in the name service database. The entry name of the group is required. Unless it is limited to a specific member (by the \texttt{-m} option), the \texttt{show group} command shows all members. The command shows only the members in the specified group; the \texttt{-r} option enables you to show members of nested groups.

Privilege Required: You must have read permission to the CDS object entry (the target group entry). If you use the \texttt{-r} option, you also require read permission to any nested groups.

Note: The \texttt{show group} command is supported in z/OS DCE by the \texttt{dcecp rcpgroup show} command.
Examples

The following example shows all the members of the group `/.:LandS/anthro/Calendar_group`, in the order in which they were added to the group:

```
READY
    rpccp
rpccp>  show group /.:LandS/anthro/Calendar_group
```

The following command operates from the system prompt to show a specific member of the group `/.:LandS/dept/Calendar_group`:

```
READY
    rpccp rpccp>  show group  \
    >  -m /.:LandS/anthro/Cal_host_2  \
    >  /.:LandS/anthro/Calendar_group
```

Related Information

Commands:

```
add member    remove group    remove member
```
show mapping

Shows the elements of either a local endpoint map or a remote endpoint map.

Format

rpccp show mapping [host-address]
[ -i if-id [ -v versions]]
[ -o object-uuid]

Options

- **-i** Defines an interface identifier to be shown (optional). Only one interface can be shown in a single operation. If specified, only elements containing this interface identifier are shown. The -i option can be qualified by the -v option.

The interface identifier value has the following form:

```
interface-uuid[ major-version.minor-version]
```

`major-version.minor-version` is optional; defaults to 0,0.

The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:

```
- i  EC1EEB60-5943-11C9-A309-08002B102989,1.1
- i  ec1eeb60-5943-11c9-a309-08002b102989,1.1
```

Leading zeros in version numbers are ignored.

- **-v** Indicates how a specified interface version is used (optional). If it is used without the -i option, the -v option is ignored. The possible combinations of versions for the -v option and their actions are as follows:

<table>
<thead>
<tr>
<th>Versions</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>This interface version is ignored.</td>
</tr>
<tr>
<td>exact</td>
<td>Both the major and minor versions must match the specified versions.</td>
</tr>
<tr>
<td>compatible</td>
<td>The major version must match the specified version, and the minor version must be greater than or equal to the specified version.</td>
</tr>
<tr>
<td>major_only</td>
<td>The major version must match the specified version; the minor version is ignored.</td>
</tr>
<tr>
<td>upto</td>
<td>The major version must be less than or equal to that specified. If the major versions are equal, the minor version must be less than or equal to that specified.</td>
</tr>
</tbody>
</table>

If the -v option is absent, the command shows compatible version numbers.

- **-o** Defines an object to be shown (optional).

The UUID is a hexadecimal string, for example:

```
- o  3c6b8f60-5945-11c9-a236-08002b102989
```

**Note:** The options list a selected subset of map elements. The -i option selects a specific interface, and the -v option qualifies the -i option. The -o option selects a specific object. The options work together to specify the subset of elements for the target protocol sequence, or sequences.
Parameters

host-address

This parameter is a string binding that indicates where to find the target endpoint map. When accessing the local endpoint map, you can specify which protocol sequence to use (optional), for example:

ncadg_ip_udp:

When accessing a remote endpoint map, you must specify both a protocol sequence and a network address for the remote system (required), for example:

ncadg_ip_udp:16.20.16.44

An endpoint is unnecessary in local or remote host addresses, and the show mapping command ignores any endpoint specified as part of a host address.

Usage

The show mapping command shows elements of an endpoint map. Each element corresponds to an object UUID, interface identifier, annotation, and binding information. The binding information contains an RPC protocol sequence, a network address, and an endpoint within square brackets. For example, rpc-prot-seq:network-addr[endpoint].

The endpoint map can be either the local endpoint map or the endpoint map of a specified remote host. If entered without a remote host address, the command accesses the local endpoint map. For the local endpoint map, a show mapping command without any options displays all the map elements. For a remote endpoint map, map elements are accessible only for protocol sequences that are supported on both your system and the remote system.

Note: To ensure that you can remotely display all map elements from every remote endpoint map, run the RPC control program on a system that supports all of the protocol sequences available in your network environment.

Note: The show mapping command is supported in z/OS DCE by the dcecp endpoint show command.

Examples

The following commands start the control program and show the map elements in the local endpoint map that contain the specified interface identifier:

READY
rpccp
rpccp> show mapping -i ec1eeb60-5943-11c9-a309-08002b102989,1.1

The following rpccp show mapping command operates from the system prompt. The command accesses the endpoint map of the remote host specified by the host address (ncadg_ip_udp:16.20.16.44), and displays the one map element that contains both the specified interface identifier and the specified object UUID:

READY
rpccp show mapping \
rpccp> -i ec1eeb60-5943-11c9-a309-08002b102989,1.1 \
> -o 30dbeea0-fb6c-11c9-8eaa-08002b0f4528 \
> ncadg_ip_udp:16.20.16.44
show mapping

Related Information

Commands:

remove mapping    show server
show profile

Shows the elements of a profile.

Format

rpccp show profile  profile-entry-name
{-d | -a annotation | -i if-id [-v versions] -m member}
[-r [ integer]] [-s syntax] [-u]

Options

-d  Selects the default profile element. With the -d option, the -a, -i, and -m options are ignored.

Note that although the -a option will work with the -d option, they are not meant to be used together.

-a  Declares a single annotation field (optional). The -a option selects only elements containing the specified annotation; it is case sensitive.

The -a option works alone or in combination with the -i or -m options or both; only elements containing all the specified values are displayed.

Note that the shell supports quotation marks around the annotation field of profile elements, to allow you to include internal spaces in an annotation. The control program does not support quotation marks around the annotation field. To specify or refer to annotations from within the control program, limit each annotation to an unbroken alphanumeric string, for example, CalendarGroup. To refer to annotations from the system prompt, do not incorporate quotation marks into any annotation.

-i  Selects a specified interface identifier (optional). Only elements containing that interface identifier are shown. The interface identifier value has the following form:

interface-uuid[major-version.minor-version]

major-version.minor-version is optional; defaults to 0,0.

The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:

-i ec1eeb60-5943-11c9-a309-08002b102989,1.1

Leading zeros in version numbers are ignored.

The -i option works alone or in combination with the -a or -m options or both; only elements containing all the specified values are displayed. When the -d option is specified, the -i option is ignored.

-v  Indicates how a specified interface version is used (optional). If it is used without the -i option, the -v option is ignored. The possible combinations of versions for the -v option and their actions are described in the following table:

<table>
<thead>
<tr>
<th>Versions</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>This interface version is ignored.</td>
</tr>
<tr>
<td>exact</td>
<td>Both the major and minor versions must match the specified versions.</td>
</tr>
</tbody>
</table>
show profile

compatible  The major version must match the specified version, and the minor version must be greater than or equal to the specified version.

major_only  The major version must match the specified version; the minor version is ignored.

upto  The major version must be less than or equal to that specified. If the major versions are equal, the minor version must be less than or equal to that specified.

If the -v option is absent, the command shows compatible version numbers.

-m  Declares a single member name (optional). Only elements containing that member name are shown.

The -m option works alone or in combination with the -a or -i options or both; only elements containing all the specified values are displayed. When the -d option is specified, the -m option is ignored.

-r  Indicates that the show profile operation is recursive. If the member of any element of a profile is also a profile, its elements are shown. By default, the -r option causes the show profile operation to be called until all nested profiles are expanded. For example, -r shows the elements of the specified profile and of all nested profiles.

You can limit recursion to one or more levels by specifying a decimal integer as part of the -r option. For example, -r 1 shows the elements of the specified profile and, for element members that are profiles, the command also shows their elements; then recursion stops.

Without the -r option, only the profile elements in the specified entry are shown.

-s  Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

-u  Updates the local CDS cache copy of name service data (optional).

Name service data is cached locally on each machine in a cell. If an rpccp inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the required data is not available in the local CDS cache, rpccp goes to a CDS server(s) to retrieve the required data. Then rpccp updates the local CDS cache.

Using the -u option bypasses the local cache, allowing rpccp to go directly to a CDS server for the inquiry. Then rpccp updates the local CDS cache.

Parameters

profile-entry-name  Indicates the name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The show profile command shows the elements of a profile in the name service database. The entry name of the profile is required.

By default, all elements in the profile are shown. You can select a subset of the elements by specifying the -a, -i, or -m options. The -r option enables you to show nested profiles.
Privilege Required: You must have read permission to the CDS object entry (the target profile entry). If you use the -r option, you also require read permission to any nested profiles.

Note: The show profile command is supported in z/OS DCE by the dcecp rpcprofile show command.

Examples

The following command operates from the TSO command line to show the cell profile /./cell-profile in the local cell:

rpccp show profile /./cell-profile

Related Information

Commands:

add element       remove element       remove profile
show server

Shows the binding information, interface identifiers, and object UUIDs in a server entry.

Format

rpccp show server server-entry-name
   [-i [if-id]]
   [-o [object-uuid]]
   [-s syntax] [-u]

Options

- **-i**
  Shows interface identifiers from binding information found in the entry (optional). Without the -i option, the command displays all interface identifiers.
  To display a specific interface, supply its identifier as the value. The value has the following form:
  
  interface-uuid[major-version.minor-version]
  
  major-version.minor-version is optional; defaults to 0,0.
  The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:
  
  -i ec1eeb60-5943-11c9-a309-08002b102989,1.1
  
  Leading zeros in version numbers are ignored.

- **-o**
  Shows object UUIDs found in the entry (optional). Without the -o option, the command displays all object UUIDs. To display a specific object UUID, supply its string representation as the value, for example:
  
  -o 3c6b8f60-5945-11c9-a236-08002b102989

- **-s**
  Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Specifying the -s option is unnecessary.

- **-u**
  Updates the local CDS cache copy of name service data (optional).
  Name service data is cached locally on each machine in a cell. If an rpccp inquiry can be satisfied by data in the local CDS cache, this cached data is returned. However, locally cached copies of name service data might not include a recent CDS update. If the required data is not available in the local CDS cache, rpccp goes to a CDS server(s) to retrieve the required data. Then rpccp updates the local CDS cache.
  
  Using the -u option bypasses the local cache, allowing rpccp to go directly to a CDS server for the inquiry. Then rpccp updates the local CDS cache.

Parameters

**server-entry-name**

Indicates the name of the target server. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.
Usage

The `show server` command shows the RPC binding information, interface identifiers, and object UUIDs in a server entry. The entry name of the server entry is required.

This operation shows all of the potential bindings for an interface. By default, this command displays bindings for the specified version of the interface and for upwardly compatible versions of the interface.

Privilege Required: You must have read permission to the CDS object entry (the target server entry).

Note: The `show server` command is supported in z/OS DCE by the `dcecp rpcentry show` command.

Examples

The following commands start the control program and show the server entry `/.:/LandS/anthro/Cal_host_2` in the local cell:

```
READY
   rpccp
rpccp> show server
./:/LandS/anthro/Cal_host_2
```

The following command starts the control program and then displays a specific object and interface from the server entry `/.:/LandS/anthro/Cal_host_2` in the local cell:

```
READY
   rpccp
rpccp> show server
> ./:/LandS/anthro/Cal_host_2 \ 
> -o 16977538-E257-11C9-8DCwzerodot-wzerodot8wzerodotwzerodot2BwzerodotF4528 \ 
> -i EC1EEB6wzerodot-5943-11C9-A3wzerodot9-wzerodot8wzerodotwzerodot2B1wzerodot2989,1.1 
> -o 16977538-e257-11c9-8dc0-08002b0f4528 \ 
> -i ec1eeb6wzerodot-5943-11c9-a309-08002b102989,1.1
```

Related Information

Commands:

- `export`
- `import`
- `unexport`
unexport

Removes binding information, interface identifiers, and object UUIDs from a server entry.

Format

```
rpccp unexport entry-name
([[-i if-id [-v versions]] | [-o object-uuid]])
[-s syntax]
```

Options

- **-i**  
  Defines an interface identifier to be unexported (optional). Only one interface can be unexported in a single operation. If specified, binding information for this interface is removed from the entry. The **-i** option can be qualified by the **-v** option.
  
  The interface identifier value has the following form:
  ```
  interface-uuid,major-version.minor-version
  ```
  
  The UUID is a hexadecimal string, and the version numbers are decimal strings, for example:
  ```
  -i ec1eeb60-5943-11c9-a309-08002b102989,1.1
  ```
  
  Leading zeros in version numbers are ignored.

- **-v**  
  Indicates how a specified interface version is used (optional). If it is used without the **-i** option, the **-v** option is ignored. The possible combinations of versions for the **-v** option and their actions are described in the following table:

<table>
<thead>
<tr>
<th>Versions</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>This interface version is ignored.</td>
</tr>
<tr>
<td>exact</td>
<td>Both the major and minor versions must match the specified versions.</td>
</tr>
<tr>
<td>compatible</td>
<td>The major version must match the specified version, and the minor version must be greater than or equal to the specified version.</td>
</tr>
<tr>
<td>major_only</td>
<td>The major version must match the specified version; the minor version is ignored.</td>
</tr>
<tr>
<td>upto</td>
<td>The major version must be less than or equal to that specified. If the major versions are equal, the minor version must be less than or equal to that specified.</td>
</tr>
</tbody>
</table>

  If the **-v** option is absent, the command shows compatible version numbers.

- **-o**  
  Defines an object to be unexported (optional). Each **unexport** command accepts up to 32 **-o** options.
  
  The UUID is a hexadecimal string, for example:
  ```
  -o 3c6b8f60-5945-11c9-a236-08002b102989
  ```

- **-s**  
  Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Specifying the **-s** option is unnecessary.

Parameters
entry-name Indicates the name of the target name service entry. Usually, the target is a server entry. However, objects also can be exported (without an interface identifier or binding information) to a group or a profile.

For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Usage

The unexport command removes binding information and an interface identifier, object UUIDs, or both from a server entry, or it removes object UUIDs from a group’s entry. The command requires the entry name and either the interface identifier or one or more object UUIDs.

By default, the unexport operation removes compatible interface versions.

Privilege Required: You must have both read permission and write permission to the CDS object entry (the target name service entry).

Note: The unexport command is supported in z/OS DCE by the dcecp rpcentry unexport command.

Examples

The initial commands set up an environment variable Calendar_1_1, representing the interface identified of an RPC interface.

The control program commands start the control program and remove (unexport) the interface from the server entry ./LandS/anthro/Cal_host_2 in the local cell:

READY
$ setenv Calendar_1_1 ec1eeb6wzerodot-5943-11c9-a309-08002b102989,1.1
$ export Calendar_1_1
READY
rpccp> unexport \n
> -i export Calendar_1_1 \n
> ./LandS/anthro/Cal_host_2
rpccp>

Related Information

Commands:

export import show server
uuidgen

Purpose
Generates a universal unique identifier (UUID).

Format
uuidgen [option]...

Options
-i
Produces an interface definition language (IDL) file template and includes the generated UUID string in the template. If both the -i and -s options are specified, the last one specified in the uuidgen command is used and the other is ignored.

-o file name
Redirects the generated UUID string to the file you specify.

Note: When UUIDGEN is used in the TSO/E environment, the file name must specify a partition data set name with its member name in parentheses ( ). When this option is used in the JCL environment, the file name following the -o refers to the DD statement containing the name of the partition data set and member name. See Examples for details.

-s
Generates a UUID string as an initialized C structure. If both the -i and -s options are specified, the last one specified in the uuidgen command is used and the other is ignored.

-t old_style_uuid_string
Translates the old-style (NCS Version 1) UUID string format to the new style string format.

The -t option must be specified as the last option in the uuidgen command; all options that follow the -t option are ignored. The -i, -s, and -n options are ignored even when specified before the -t

-v
Displays information about the UUID generator but does not generate a UUID.

-h
Displays information about the uuidgen command options. The options -h and -? can be used interchangeably.

-?
Displays information about the uuidgen command options. The options -? and -h can be used interchangeably.

-n number_of_uuid_strings
Generates a specified number of UUID strings.

-c
Lets you supply an existing UUID that uuidgen then outputs in the format you specify. This option is especially useful in combination with the -s or -i options for converting an existing UUID into a C-structure or an IDL file template.

Note:
The -c option must be specified at the end of the uuidgen command; all options that follow the -c option are ignored. The -n option is ignored even when specified before the -c option in the uuidgen command.
Usage

The `uuidgen` command creates a UUID string that you assign to an object to uniquely identify it. One such use is in the UUID interface attribute of an IDL interface definition. The format for representing a UUID string consists of eight hexadecimal digits followed by a hyphen, followed by three groups of four hexadecimal digits separated by hyphens, followed by a hyphen and twelve hexadecimal digits.

When the `-v`, `-h`, and `-?` options are used in the `uuid` command, all other options are ignored. If the `-v`, `-h`, and `-?` options are all specified, the first one specified in the `uuidgen` command is processed, and the others are ignored.

Restriction

The z/OS DCE UUIDGEN command must be run in code page IBM-1047. See the z/OS DCE Administration Guide for more information.

Examples

1. The following example generates a UUID string:

   ```
   READY
   uuidgen
   23c67e00-71b6-11c9-9dfc-8002b0eece
   ```

2. The following example generates a partial template, containing a generated UUID string, used to develop an interface definition.

   ```
   //JOBNAME JOB (ACCOUNT), ... your job parameters
   //**************
   /// Note: The partition data set, usurprfx.yyyyyyy.idl must be
   /// pre-allocated. Member "myidl" will be created in this
   /// PDS.
   //UUIDGEN EXEC UUIDGEN,
   // PARMS='-i -o //DD:UUIDOUT'
   //UUIDOUT DD DSN=USRPRFX.YYYYYY.IDL(MYIDL),DISP=SHR
   ///
   ```

   In the JCL, you can specify the options as PARMS of the JCL. The data set that holds the output IDL file should be pre-allocated. In this example, an IDL template file will be placed in data set `usrprfx.yyyyyyy.idl` using the member name "myname".

3. The following example converts a UUID string from the old style format to the new format.

   ```
   READY
   uuidgen -t 34DC23469EAF.AB.A2.01.7C.5F.2C.ED.A3
   34dc2346-9eaf-0000-aba2-017c5f2ceda3
   ```

4. The following example is running `uuidgen` under TSO/E.

   ```
   READY
   uuidgen -i -o "//'userprfx.yyy.idl(mbrname)"
   ```

   where `userprfx.yyy.idl` specifies a partition data set and the `mbrname` is the member to be created by `uuidgen`.

For more examples, including running UUIDGEN from Batch, see the z/OS DCE Application Development Guide: Introduction and Style.
Chapter 4. Cell Directory Service

This chapter introduces you to the DCE Cell Directory Service (CDS), and describes the CDS control program administrative facility and its associated commands.

For information on starting the CDS daemons, see "Cell Directory Service Daemons" on page 508.

The DCE Cell Directory Service provides an interactive management program to manage the CDS name space contents. The cdscp commands are entered on the command line to start these management services.

Examples in this chapter use cell-relative names for entries in the CDS name space. Global names are also used in these examples. When a command is issued with a global name, the only difference is the path name.

In the example that follows, the path name in the cell-relative name is /.:, while the path name in the global name is /.../cell1. The path name /.: is a short form notation of the long form /.../cell1, where cell1 is the local cell name.

When reading the examples in this section, keep in mind that they are written to access entries in the local cell. For more information about cell-relative and global names, refer to the z/OS DCE Administration Guide.

The following example uses a cell-relative name and a global name with the add directory command to add an attribute to a directory:

1. Cell-relative name

   Enter the following command using a cell-relative name to assign the value ontario to the attribute myname of the directory sales in cell1:

   cdscp> add directory /.:/sales myname = ontario

2. Global name

   Enter the following command using a global name to assign the value ontario to the attribute myname of the directory sales in cell1:

   cdscp> add directory /.../cell1/sales myname = ontario

Related Information

Commands:

cdscp
Cell Directory Service Daemons

Four CDS daemons are supported in z/OS DCE: the CDS advertiser daemon (cdsadv), the CDS clerk daemon (cdsclerk), the CDS daemon (cdsd), and the Global Directory Agent daemon (gdad). You can start, stop, and view the status of the CDS daemon using the modify system operator command. For further information about the modify command, see Chapter 1, “DCE and z/OS Commands” on page 1.
Cell Directory Service Administrative Utilities

The CDS administration utilities enable intercell communication. Intercell communication allows DCE principles in one cell the access to DCE objects in a foreign cell.

To enable intercell communication, a DCE administrator must define both cells for which intercell communication is being established to a global naming service (X.500 or DNS). To assist the DCE administrator in defining the DCE cell to the global naming service, two CDS administration utilities are provided:

**ldap_addcell utility**
Generates CDS cell information to enable intercell communication between two DCE cells using the **LDAP** conduit in the CDS Global Directory Agent.

**mkdceregister utility**
Generates CDS cell information to enable intercell communication between two DCE cells using the **BIND** conduit in the CDS Global Directory Agent.

Determining which utility to use depends on the name syntax of the DCE cell being defined to the global naming service:

- Use the **mkdceregister** administrative utility to define a DCE Cell, whose name is untyped (DNS syntax), to the DNS global naming service.
- Use the **ldap_addcell** administrative utility to define a DCE Cell, whose name is typed (X.500 syntax), to a global naming service which supports the LDAP protocol.

Follow the procedure outlined in the [z/OS DCE Administration Guide](#) for setting up intercell communication.
ldap_addcell

Generates CDS cell information for intercell communications.

Format

```
ldap_addcell -h address [:port] -a auth_dn -p auth_dn_pw [-d]
```

Parameters

- **-h** Specifies the address of the LDAP server. This can be specified as a TCP/IP host name or as an IP address.
  The :port specification, if not specified, defaults to 389.

- **-a** Specifies the Distinguished Name (DN), set up by the LDAP administrator, from which authentication information is obtained when binding to the LDAP server.
  LDAP_AUTH_SIMPLE level authentication sets up the connection to the LDAP server.

- **-p** The password used for LDAP authentication, set up by the LDAP administrator.

- **-d** Specifies to delete the CDS_CELL and CDS_REPLICAS attribute information for this cell, rather than storing updated information. When used, this optional parameter overrides the other options.

Usage

Use the `ldap_addcell` administrative utility to generate CDS cell information to enable intercell communication between two DCE cells. The `ldap_addcell` utility generates data and registers the local cell in the LDAP server specified on the command. The CDS Cell information generated is automatically stored in the specified LDAP Server at the Domain Name which maps to the CDS cell name, in the CDS_CELL (single_valued) and CDS_REPLICAS (multi-valued) attributes. After the data is generated, the administrator must run the `dcecp registry connect` command before intercell communications can take place.

For example:

```
cellname /.../C=US/O=IBM/CN=DCECELL1
```

maps to DN

```
CN=DCECELL1, O=IBM, C=US
```

in the LDAP server.

- The user issuing this utility must be logged into DCE before running the utility.
- The user running this utility must have root userid privileges; for example, **UID=0**.

To resolve host names into IP addresses, the environment variable, **RESOLVER_CONFIG**, must be set before running the `ldap_addcell` utility. Its value must be set to the TCP/IP data set containing the TCP/IP client system parameters; for example, **hlq.TCPIP.DATA**.

After the data is generated, the administrator must run the `dcecp registry connect` command.

For more information on the `ldap_addcell` utility, see the section on Managing Intercell Naming in the

**Z/OS DCE Administration Guide**
ldap_addcell

Note:

You only need to run this utility if you are planning to enable the LDAP conduit in the CDS Global Directory Agent. You must run this utility before the LDAP conduit can successfully service requests consisting of global names whose cell name portion is typed (X.500 syntax).

Environment variables and their association to command line options:

**LDAP_SERVER** Equivalent to the `-h` command line option and is set to the host name or IP address of the LDAP Server.

**LDAP_AUTH_DN** Equivalent to the `-a` command line option and is set to the Distinguished Name from which authentication is obtained.

**LDAP_AUTH_DN_PW** Equivalent to the `-p` command line option and is set to the password used for LDAP authentication.

Examples

The following is an example of using the the `ldap_addcell` utility for a cell named /.../C=US/O=IBM/CN=DCECELL_SAMPLE:

```
> echo $RESOLVER_CONFIG
//'SHR.TCPIP.DATA'
> ldap_addcell -h ldapserver.endicott.ibm.com \
   -a "cn=ldap_addcell,o=IBM,c=US" \
   -p secret_pw
```

```
EUVC24631I The ldap_addcell utility started.
EUVC24632I Attempting to add DCE Cell information to attributes at Distinguished Name: CN=DCECELL_SAMPLE,O=IBM,C=US.
EUVC24633I The ldap_addcell utility has stored the following information:
EUVC24634I CDS_CELL attribute value:
EUVC24635I Namespace Uuid: RzerodotRzerodotc8Rzerodot55c-b5fd-11dRzerodot-bRzerodot63-Rzerodot8RzerodotRzerodot5a1916c
EUVC24636I CDS_REPLICAS attribute value:
EUVC24635I Clearinghouse Uuid: RzerodotRzerodot32efe4-b5fd-11dRzerodot-bRzerodot63-Rzerodot8RzerodotRzerodot5191a6c
EUVC24635I Clearinghouse Name: /.../dcecell14.endicott.ibm.cm/dcecell14_ch
EUVC24635I Tower: ncacn_ip_tcp:9.13Rzerodot.44.5Rzerodot{}
EUVC24635I Tower: ncadg_ip_udp:9.13Rzerodot.44.5Rzerodot{}
EUVC24636I CDS_REPLICAS attribute value:
EUVC24635I Clearinghouse Uuid: e9dd8cRzerodot1-3b34-1ecRzerodot-ad71-Rzerodot8RzerodotRzerodot53114d5
EUVC24635I Clearinghouse Name: /.../dcecell14.endicott.ibm.cm/DCECDS3_ch
EUVC24635I Tower: ncacn_ip_tcp:9.130.79.39[]
EUVC24635I Tower: ncadg_ip_udp:9.130.79.39[]
EUVC24636I CDS_REPLICAS attribute value:
EUVC24635I Clearinghouse Uuid: e9dd8c01-3b34-1ec0-ad71-080053114d5
EUVC24635I Clearinghouse Name: /.../dcecell14.endicott.ibm.cm/DCECDS3_ch
EUVC24635I Tower: ncacn_ip_tcp:9.130.79.39[]
EUVC24635I Tower: ncadg_ip_udp:9.130.79.39[]
EUVC24640I The ldap_addcell utility completed successfully.
```

Related Information

Books:

- [z/OS DCE Administration Guide](#)
**mkdceregister**

Generates CDS cell information for intercell communications.

**Format**

```
mkdceregister -o file name -t file name [-d]
```

**Parameters**

- `-o`  
  Specifies HFS output data file that stores the results of the `mkdceregister` utility.

- `-t`  
  Specifies the temporary HFS file name that stores `mkdceregister` utility processing data.

- `-d`  
  Specifies to delete the temporary HFS file name when `mkdceregister` utility processing completes.

**Usage**

Use the `mkdceregister` administrative utility to generate CDS cell information to enable intercell communication between two DCE cells. The `mkdceregister` utility generates data needed to register the local cell in the TCP/IP Domain Name Server (DNS). After the data is generated, the administrator must enter the data into the external database, and run the `dcecp registry connect` command before intercell communications can take place.

- The user issuing this utility must be logged into DCE before running the utility.
- Data generated is in the format of DNS type TXT Resource Records.
- No special privileges are required to run the utility.

The environment variables for NLSPATH and LANG must be set to the appropriate locale. This lets you receive error messages from the utility.

This utility must be run, and the subsequent data entered into the DNS database, before intercell communication can be accomplished using the BIND conduit of the CDS Global Directory Agent Daemon (GDAD).

For more information on the `mkdceregister` utility, see the section on Managing Intercell Naming in the [Z/OS DCE Administration Guide](https://www-01.ibm.com/support/knowledgecenter/SS7H96_9.0.0/com.ibm.zos.r9e.9.0.doc/SSH000010120.html).

**Note:** You only need to run this utility if you are planning to enable the BIND conduit in the CDS Global Directory Agent. You must run this utility before the BIND conduit can successfully service requests consisting of global names whose cell name portion is untyped (DNS syntax).

**Examples**

The following is an example of using the `mkdceregister` utility:

```
mkdceregister -o /tmp/mkdceregister.out -t /tmp/mkdceregister tmp
```
mkdcregister

> echo $LANG $NLSPATH
En_US.IBM-1047 /usr/lib/nls/msg/%L/%N

> mkdcregister -t /tmp/celldata.tmp -o /tmp/celldata.out
The mkdcregister utility is starting.
The mkdcregister utility is exiting with rc = 0.

> cat /tmp/celldata.out

;BEGIN DCE CELL /.../dcecell14.endicott.ibm.com INFORMATION
;Primary CDS server
dcecell14.endicott.ibm.com. IN MX 1 dcehost
dcecell14.endicott.ibm.com. IN A 9.130.44.50
;END DCE CELL /.../dcecell14.endicott.ibm.com INFORMATION

Related Information

Books:

- [z/OS DCE Administration Guide](#)
add directory

Adds a value to a modifiable, set-valued attribute (including application-defined attributes) of a directory.

Format

cdscp a[dd] [directo]y directory-name attribute-name [=] attribute-value

Parameters

directory-name  The full name of the directory.
attribute-name  The name of a particular attribute. Specify only one attribute at a time. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes that your application uses.
attribute-value  The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in “Attributes” on page 525.

Usage

The add directory command adds a value to a modifiable, set-valued (for example, multi-valued) attribute (including application-defined attributes) of a directory. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. See the z/OS DCE Administration Guide for more information about attributes.

Privilege Required:  You must have w (write) permission to the directory.

Note:  The add directory command is supported in z/OS DCE by the dcecp directory add command.

Examples

To add the value ontario to the attribute myname of a directory named /./sales, read the cds_attributes file to verify that the attribute shown in the following display exists:

<table>
<thead>
<tr>
<th>OID</th>
<th>LABEL</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.22.1.3.91</td>
<td>myname</td>
<td>char</td>
</tr>
</tbody>
</table>

Enter the following command to assign the value ontario to the attribute myname:

cdscp> add directory /./sales myname = ontario

Related Information

Commands:

create directory  list directory  set directory
delete directory  remove directory  show directory
add directory

Books:

- *z/OS DCE Administration Guide*
add object

Adds a value to a modifiable, set-valued attribute (including application-defined attributes) of an object entry.

Format

cdscp a[dd] o[bject] object-name attribute-name [=] attribute-value

Parameters

object-name The full name of the object entry.
attribute-name The name of a particular attribute. Specify only one attribute at a time. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes and corresponding data types that your application uses.
attribute-value The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in “Attributes” on page 525.

Usage

The add object command adds a value to a modifiable, set-valued (for example, multi-valued) attribute (including application-defined attributes) of an object entry. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. See the Directory Service section in the z/OS DCE Administration Guide for more information about attributes.

Privilege Required: You must have w (write) permission to the object entry.
Note: The add object command is supported in z/OS DCE by the dcecp object modify command.

Examples

To add the value ps to the attribute printcap of an object entry named /./subsys/deskprinter, read the cds_attributes file to verify that the attribute shown in the following display exists:

<table>
<thead>
<tr>
<th>OID</th>
<th>LABEL</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.22.1.3.70</td>
<td>printcap</td>
<td>char</td>
</tr>
</tbody>
</table>

Enter the following command to assign the value ps to the attribute printcap:

cdscp> add object ././subsys/deskprinter printcap = ps

Related Information

Commands:
create object list object set object
delete object remove object show object

Books:

- z/OS DCE Administration Guide
cdscp

Starts the cds control program.

CDS Control Program Commands

The CDS control program commands are:

Table 12 (Page 1 of 2). CDSCP Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add directory</td>
<td>Adds a value to a modifiable, set-valued (for example, multi-valued) attribute of a directory</td>
</tr>
<tr>
<td>add object</td>
<td>Adds a value to a modifiable, set-valued attribute of an object</td>
</tr>
<tr>
<td>clear cached server</td>
<td>Removes knowledge of a specifically defined server from the local clerk's cache</td>
</tr>
<tr>
<td>clear clearinghouse</td>
<td>Removes knowledge of the specified clearinghouse from the memory of the server</td>
</tr>
<tr>
<td>create child</td>
<td>Creates a child pointer in the master replica of the parent directory</td>
</tr>
<tr>
<td>create clearinghouse</td>
<td>Creates a clearinghouse on the local server system, or makes an existing clearinghouse available</td>
</tr>
<tr>
<td>create directory</td>
<td>Creates a directory</td>
</tr>
<tr>
<td>create link</td>
<td>Creates a soft link and optionally specifies an expiration time and extension time</td>
</tr>
<tr>
<td>create object</td>
<td>Creates an object entry</td>
</tr>
<tr>
<td>create replica</td>
<td>Creates a replica of an existing directory in the specified clearinghouse</td>
</tr>
<tr>
<td>define cached server</td>
<td>Creates knowledge of a server in the local clerk's cache</td>
</tr>
<tr>
<td>delete child</td>
<td>Deletes a child pointer from the name space</td>
</tr>
<tr>
<td>delete clearinghouse</td>
<td>Deletes the specified clearinghouse from the local server system</td>
</tr>
<tr>
<td>delete directory</td>
<td>Deletes a directory</td>
</tr>
<tr>
<td>delete link</td>
<td>Deletes a soft link</td>
</tr>
<tr>
<td>delete object</td>
<td>Deletes an object entry</td>
</tr>
<tr>
<td>delete replica</td>
<td>Deletes a read-only replica of a directory from a clearinghouse</td>
</tr>
<tr>
<td>disable clerk</td>
<td>Stops the clerk on the local system</td>
</tr>
<tr>
<td>disable server</td>
<td>Stops the server on the local system</td>
</tr>
<tr>
<td>do filename</td>
<td>Runs cdscp commands contained in a specified file</td>
</tr>
<tr>
<td>dump clerk cache</td>
<td>Displays the contents of the clerk cache</td>
</tr>
<tr>
<td>help</td>
<td>Displays a list of available cdscp commands</td>
</tr>
<tr>
<td>list child</td>
<td>Displays a list of all of the child pointers whose names match the specified child name</td>
</tr>
<tr>
<td>list clearinghouse</td>
<td>Displays a list of all of the clearinghouses whose names match the specified clearinghouse name</td>
</tr>
<tr>
<td>list directory</td>
<td>Displays a list of all of the directories whose names match the specified directory name</td>
</tr>
<tr>
<td>list link</td>
<td>Displays a list of all of the soft links whose names match the link name specified</td>
</tr>
</tbody>
</table>
### Table 12 (Page 2 of 2). CDSCP Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list object</td>
<td>Lists the specific object entries whose names match the object entry name specified</td>
</tr>
<tr>
<td>remove directory</td>
<td>Removes a value from a set-valued attribute of a directory</td>
</tr>
<tr>
<td>remove link</td>
<td>Removes the timeout value attribute of a soft link</td>
</tr>
<tr>
<td>remove object</td>
<td>Removes a value from a set-valued or single-valued attribute of an object entry</td>
</tr>
<tr>
<td>set cdscp confidence</td>
<td>Sets the confidence level of cdscp</td>
</tr>
<tr>
<td>set cdscp preferred clearinghouse</td>
<td>Specifies a preferred clearinghouse to use for satisfying read requests that result from cdscp commands</td>
</tr>
<tr>
<td>set directory</td>
<td>Changes the value of a modifiable, single-valued attribute of a directory</td>
</tr>
<tr>
<td>set directory to new epoch</td>
<td>Rebuilds a directory replica set and allows to designate a new master replica or to exclude a replica</td>
</tr>
<tr>
<td>set directory to skulk</td>
<td>Immediately starts the skulk of a directory</td>
</tr>
<tr>
<td>set link</td>
<td>Changes the value of the modifiable, single-valued attributes of a soft link</td>
</tr>
<tr>
<td>set object</td>
<td>Changes the value of a modifiable, single-valued attribute of an object entry</td>
</tr>
<tr>
<td>show cached clearinghouse</td>
<td>Displays current information about the specified clearinghouse cached in the CDS clerk used by cdscp commands</td>
</tr>
<tr>
<td>show cached server</td>
<td>Displays address information of a server in the CDS clerk’s cache</td>
</tr>
<tr>
<td>show cdscp confidence</td>
<td>Displays the current confidence level of clerk calls resulting from cdscp commands</td>
</tr>
<tr>
<td>show cdscp preferred clearinghouse</td>
<td>Displays the preferred clearinghouse for satisfying read requests that result from cdscp commands</td>
</tr>
<tr>
<td>show cdscp syntax</td>
<td>Displays the syntax of the available cdscp commands.</td>
</tr>
<tr>
<td>show cell</td>
<td>Displays the information required to create a cell entry in either the DNS or GDS</td>
</tr>
<tr>
<td>show child</td>
<td>Displays attribute information about the specified child pointer</td>
</tr>
<tr>
<td>show clearinghouse</td>
<td>Displays attribute information about the specified clearinghouse</td>
</tr>
<tr>
<td>show clerk</td>
<td>Displays attribute information about the CDS clerk on the local serve</td>
</tr>
<tr>
<td>show directory</td>
<td>Displays attribute information about the specified directory</td>
</tr>
<tr>
<td>show link</td>
<td>Displays attribute information about the specified soft link</td>
</tr>
<tr>
<td>show object</td>
<td>Displays attribute information about the specified object entry</td>
</tr>
<tr>
<td>show replica</td>
<td>Displays attribute information about the specified replica</td>
</tr>
<tr>
<td>show server</td>
<td>Displays attribute information about the server running on the local system</td>
</tr>
</tbody>
</table>

For more information on any of the above cdscp commands, see the appropriate page in this chapter.

**Note:**

With the exception of the following subcommands, the cdscp command is supported by the dcecp command.

- disable clerk
- set directory to new epoch
It is recommended that `dcecp` be used if the function is available in `dcecp`.

The following table shows the `dcecp` commands that support the `cdscp` commands:

<table>
<thead>
<tr>
<th><code>cdscp</code> Command</th>
<th><code>dcecp</code> Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cdscp</code></td>
<td><code>dcecp</code></td>
</tr>
<tr>
<td>add directory</td>
<td>directory add</td>
</tr>
<tr>
<td>add object</td>
<td>object modify</td>
</tr>
<tr>
<td>clear cached server</td>
<td>cdscache delete</td>
</tr>
<tr>
<td>clear clearinghouse</td>
<td>clearinghouse disable</td>
</tr>
<tr>
<td>create child</td>
<td>directory add</td>
</tr>
<tr>
<td>create clearinghouse</td>
<td>clearinghouse create</td>
</tr>
<tr>
<td>create directory</td>
<td>directory create</td>
</tr>
<tr>
<td>create link</td>
<td>link create</td>
</tr>
<tr>
<td>create object</td>
<td>object create</td>
</tr>
<tr>
<td>create replica</td>
<td>directory create</td>
</tr>
<tr>
<td>define cached server</td>
<td>cdscache create</td>
</tr>
<tr>
<td>delete child</td>
<td>directory delete</td>
</tr>
<tr>
<td>delete clearinghouse</td>
<td>clearinghouse delete</td>
</tr>
<tr>
<td>delete directory</td>
<td>directory delete</td>
</tr>
<tr>
<td>delete link</td>
<td>link delete</td>
</tr>
<tr>
<td>delete object</td>
<td>object delete</td>
</tr>
<tr>
<td>delete replica</td>
<td>directory delete</td>
</tr>
<tr>
<td>disable clerk</td>
<td>NONE</td>
</tr>
<tr>
<td>disable server</td>
<td>server disable</td>
</tr>
<tr>
<td>do <code>filename</code></td>
<td><code>source filename</code></td>
</tr>
<tr>
<td>dump clerk cache</td>
<td>cdscache dump</td>
</tr>
<tr>
<td>list child</td>
<td>directory list</td>
</tr>
<tr>
<td>list clearinghouse</td>
<td>clearinghouse catalog</td>
</tr>
<tr>
<td>list directory</td>
<td>directory list</td>
</tr>
<tr>
<td>list link</td>
<td>link show</td>
</tr>
<tr>
<td>list object</td>
<td>object show</td>
</tr>
<tr>
<td>remove directory</td>
<td>directory modify</td>
</tr>
<tr>
<td>remove link</td>
<td>link modify</td>
</tr>
<tr>
<td>remove object</td>
<td>object modify</td>
</tr>
<tr>
<td>set cdscp confidence</td>
<td>NONE</td>
</tr>
<tr>
<td>set cdscp preferred clearinghouse</td>
<td>NONE</td>
</tr>
<tr>
<td>set directory</td>
<td>directory modify</td>
</tr>
</tbody>
</table>
The Cell Directory Service (CDS) control program is a command-line interface for managing the components of CDS and the contents of the name space. You can use the control program commands from:

- Within the control program

  To use the control program commands from inside the control program, start the control program by using the `cdscp` command alone, without any arguments. This starts the control program, which displays the control-program prompt (`cdscp>`).

  READY
  cdscp
  cdscp>

  At this prompt, you can enter any control-program command. For example:

  `cdscp> show clerk`

  Use the command `do filename` from within the control program to read and process a file of commands.

  The file `$HOME/.cdscpinit` is the startup file.

  To leave the control program and return to the system prompt, use the `quit` or `exit` command.

- The system prompt

  To use the control program commands from the system prompt, enter the `cdscp` command with an internal command of the CDS control program as the argument. The control program runs the command immediately, without displaying the control-program prompt. For example, you can enter the `show clerk` command as follows:

---

**Table 13 (Page 2 of 2). DCECP Equivalent Commands for CDSCP Commands**

<table>
<thead>
<tr>
<th>cdscp Command</th>
<th>dcecp Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>set directory to new epoch</td>
<td>NONE</td>
</tr>
<tr>
<td>set directory to skulk</td>
<td>directory synchronize</td>
</tr>
<tr>
<td>set link</td>
<td>link modify</td>
</tr>
<tr>
<td>set object</td>
<td>object modify</td>
</tr>
<tr>
<td>show cached clearinghouse</td>
<td>cdscache show</td>
</tr>
<tr>
<td>show cached server</td>
<td>cdscache show</td>
</tr>
<tr>
<td>show cdscp confidence</td>
<td>NONE</td>
</tr>
<tr>
<td>show cdscp preferred clearinghouse</td>
<td>NONE</td>
</tr>
<tr>
<td>show cell</td>
<td>cell show</td>
</tr>
<tr>
<td>show child</td>
<td>directory show</td>
</tr>
<tr>
<td>show clearinghouse</td>
<td>clearinghouse show</td>
</tr>
<tr>
<td>show clerk</td>
<td>NONE</td>
</tr>
<tr>
<td>show directory</td>
<td>directory show</td>
</tr>
<tr>
<td>show link</td>
<td>link show</td>
</tr>
<tr>
<td>show object</td>
<td>object show</td>
</tr>
<tr>
<td>show replica</td>
<td>directory show</td>
</tr>
<tr>
<td>show server</td>
<td>server show</td>
</tr>
</tbody>
</table>
cdscp

READY

    cdscp show clerk

Enter all non-interactive cdscp commands from native TSO/E or the z/OS UNIX System Services shell only. If these commands and their arguments are entered from the ISPF command line, they are converted to lowercase characters. This is a problem for arguments that have mixed or all uppercase characters. For example, the following command issued from TSO actually creates the directory 

    1.:/testdir

and not the directory 1.:/TestDir.

    tso cdscp create dir 1.:/TestDir

If you run cdscp in batch mode, the input to the control programs and the output for each command are written to the job execution log. You can then determine which input commands were successful and which were not. An example of JCL used to run cdscp in batch mode follows:

522 Command Reference
//CDSUPPER JOB (),'USERA',CLASS=A,
// MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*JOBPARM L=200,TIME=1440
//*JOBLIB DD DSN=DTS.M050COL.LOAD,DISP=SHR
//CDSPROC PROC LPARM=
//GO EXEC PGM=EUVCCP,REGION=0M,TIME=1440,
// PARM=('POSIX(ON)/&LPARM')
//SYSSOUT DD DSN=USERA.HIGHRISK.CDSCH.ERR,
// DISP=(MOD,CATLG,CATLG)
//SYSSPRINT DD DSN=USERA.HIGHRISK.CDSCH,
// DISP=(MOD,CATLG,CATLG)
//CEEDUMP DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
// PEND

`---

Note: USERA.HIGHRISK.CDSCH and USERA.HIGHRISK.CDSCH.ERR
have been pre-allocated before running this job. A TSO
DCELOGIN has also been performed before running this job.
---`

`-------- SYNTAX ERROR TEST ******
ST1 EXEC PROC=CDSPROC,LPARM='cr dir error_ch'
//GO.SYSSPRINT DD DISP=(MOD,CATLG,CATLG),SPACE=(255,20),
// DCB=(RECFM=FB,LRECL=255,BLKSIZE=6120)
//GO.SYSOUT DD DISP=(MOD,CATLG,CATLG),SPACE=(255,20),
// DCB=(RECFM=FB,LRECL=255,BLKSIZE=6120)

----- VALID CDSCP TESTS ******
ST2 EXEC PROC=CDSPROC,LPARM='create dir ./:/Dir15 clear ./:/dcecx ell_ch'
//ST3 EXEC PROC=CDSPROC,LPARM='set dir ./:/Dir15 to skulk'
//ST4 EXEC PROC=CDSPROC,LPARM='list directory ./:/'*
//ST5 EXEC PROC=CDSPROC,LPARM='delete dir ./:/Dir15'
FREE FI(CLIB)

output in USERA.HIGHRISK.CDSCH output data set:
```
LIST
DIRECTORY   ...c=dcecell13
Dir15
  XPhoneBook
  greet5
  greet55
  hosts
  p2
  p3
  phone2
  phonedir
  subsys
  timop
```

output in USERA.HIGHRISK.CDSCH.ERR output data set:
```
EUVC16418E CDS control program detects syntax error.
```

Chapter 4. Cell Directory Service  523
Elements of a CDS Command: Each CDS control program command must include a verb, an entity name, and all required arguments. Depending on the command, however, optional arguments and attributes may also be specified. A space must separate these options. For optional attribute arguments, a space must be inserted between the attribute type and attribute value. All equal signs are optional, but if used, a space must be inserted before and after the equal sign. For example:

```
a = b
```
is valid, but not:
```
a=b
```

---

Important Note to Users

CDS command verbs are not case sensitive; however, CDS names and entities are case sensitive. To retrieve entries, enter CDS names in the same case in which they are stored.

---

Verbs: The following is a list of the definitions of verbs used in control-program commands:

- **add**: Adds a value to a modifiable, set-valued attribute.
- **clear**: Removes knowledge of a cached clearinghouse or cached server from memory.
- **create**: Creates an entity.
- **define**: Creates knowledge of a locally cached server.
- **delete**: Deletes an entity.
- **disable**: Stops operation of a clerk or server.
- **do**: Executes the commands in a file.
- **dump**: Displays the contents of a clerk cache to the terminal.
- **help**: Displays a list of all CDSCP commands.
- **list**: Displays a list of specified entity names to the terminal.
- **remove**: Removes a value from a set-valued or single-valued attribute.
- **set**: Changes the value of a modifiable, single-valued attribute.
- **show**: Displays attribute information to the terminal.

Entity Names: Any individually manageable piece of CDS is called an entity. A set of commands exists for each entity. The following list of entities describes what each entity represents:

- **Cached Clearinghouse**: A cached clearinghouse is an entry in the cache that contains the address of a named clearinghouse. A clerk can learn about a clearinghouse as a result of configuration information, advertisements received on a LAN, or during the process of finding a name.

- **Cached Server**: A cached server is a server that a clerk has cached as a result of manual configuration through the control program.

- **Cell**: A cell is an administrative domain.

- **Child**: A child pointer connects a parent and child directory in a hierarchical name space. The child pointer is stored in the parent directory and has the same name as the child directory.
**Clearinghouse**  A clearinghouse is a database containing a collection of directory replicas at a particular server.

**Clerk**  The clerk is the interface between client applications and servers.

**Directory**  A directory contains child, object, and link entries that are logically stored under one name (the directory name).

**Link**  A soft link is a pointer providing an alternative name for an object entry, directory, or other soft link.

**Object**  An object entry represents a resource (for example, an application) that is named in the name space.

**Replica**  A replica is a copy of a directory. Each copy, including the original or master, is referred to as a replica.

**Server**  A server handles lookup requests from clerks and maintains the contents of the clearinghouse or clearinghouses at its node.

**Attributes:**  Every CDS entity has attributes, which are pieces or sets of data associated with that entity. Attributes can reflect or affect the operational behavior of an entity, record the number of times a particular event or problem occurred since the entity was last enabled, and uniquely distinguish an entity from any other entity. Some attributes have a single value; others contain a set of values.

CDS attributes are identified by ISO object identifiers (OIDs). Every CDS attribute name maps to an OID and a corresponding data type. Usually, client applications define the name of an attribute and its data type. Except for the purpose of foreign language translation, application programmers should never have to modify the existing CDS labels associated with the unique OIDs in the `cds_attributes` file. However, programmers can obtain new OIDs from the appropriate allocation authority, create new attributes for their own object entries, and then append them to the existing list. The `cds_attributes` file, where the OID and data type of each attribute are stored, is in the `/opt/dcelocal/etc/cds_attributes` file.

**Note:**  When editing the `cds_attributes` file, do not leave any blank line or lines with unwanted characters and ensure that there is no extraneous data at the end of the file after the last attribute.

All entities have `show` commands that you can use to display the names and values of specific attributes or all attributes. When you display an attribute that has more than one value, the `show` command lists each value for the attribute separately. When there are multiple values for an attribute, the command first lists the attribute name on a line ending with a colon, then the parts of the value.

The following is a list of the attribute types and structures used in `cdscp` to add user-defined attributes to CDS entries:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>none</strong></td>
<td>Describes the absence of a value, for example NULL.</td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td>None.</td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>No value has to be assigned to declare an attribute of this type.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>TESTnone</td>
</tr>
<tr>
<td><strong>uuid</strong></td>
<td>Describes the UUID of an object.</td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td><code>uuid_t</code></td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td><code>time_low</code></td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td><code>time_mid</code></td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td><code>time_hi_and_version</code></td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td><code>clock_seq_hi_and_reversed</code></td>
</tr>
</tbody>
</table>
cdscp

unsigned8 clock_seq_low
idl_byte node[6]

Format uuid is entered as a standard UUID. Refer to the z/OS DCE Application Development Reference for details on using uuidgen to generate UUIDs.

Example TESTuuid = 018b26c-0a74-1dd1-9781-10005aa85e6a

long
Describes a signed long integer.

Structure Signed 32-bit Integer.

Format No specific format is required.

Example TESTlong = 12345678

short
Describes a signed short integer.

Structure Signed 16-bit integer.

Format No specific format is required.

Example TESTshort = 12345

small
Describes a signed small integer.

Structure Signed 8-bit integer.

Format No specific format is required.

Example TESTsmall = 123

Timestamp
Describes a unique identifier timestamp.

Structure

idl_byte ts_node[6]
idl_uhyper_int ts_time

idl_byte: mapped to byte
idl_uhyper_int: mapped to dnr_uhyper_int

Structure
	dnr_uhyper_int unsigned long integer

Format Timestamp consists of
yyyy-mm-dd-hh:mm:ss.dddd.../hx-hx-hx-hx-hx-hx where hx is a hexadecimal number. The first part of the timestamp is a date and time, and the second part is the unique node ID of the machine.

Example: TESTtmsp = 1996-05-16-21:37:18.300167100/10-00-5a-a8-5e-6a

Timeout
Contains two UTC values.

Structure

utc_t expire
utc_t extend

utc_t
Structure

idl_byte char_array[16]
Format  Timeout consists of two time parts, yyyy-mm-dd-hh:mm:ss and ddd-hh:mm:ss.

expire is an expiration date and time and extend is a period of time by which to extend the expiration time. Refer to "create link" on page 536 for details on these values.

Example  TESTTimeout = (1996-06-25-12:00:00 90-00:00:00)

Version  Describes a version number.

Structure  

unsigned char  cdsMajVer  
unsigned char  cdsMinVer  

Format  Version is represented as a two part decimal \((M,m)\), where \(M\) is the major version and \(m\) is the minor version. The value of \(M\) and \(m\) can be a positive integer from 0 - 255.

Example  TESTVersion = 3.4

char  Describes a counted string of unsigned characters.

Structure  

unsigned short  length  
char  *char_p  

Format  char can be input as a text string, where the length field of the structure is calculated automatically. A blank " " can be entered into a string by using the escape character \\.

Example  TESTchar = hello there

byte  Describes a counted string of untranslated characters.

Structure  

unsigned short  length  
Void  *byte_p  

Format  byte is entered as a string of bytes in raw hexadecimal format.

Note:  Hexadecimal characters must be entered in pairs, an even number of characters.

Example  TESTbyte = a4e5cd

ASN1  Describes data in Abstract Syntax Notation (ASN).

Structure  

unsigned short  length  
Void  *byte_p  

Format  ASN1 is entered as a string of bytes in raw hexadecimal format.

Note:  Hex characters must be entered in pairs, an even number of characters.

Example  TESTASN1 = a4e5cd

Note:  When adding an attribute to an entity and an OID is specified for the attribute name, but is not included in the cds_attributes file, the attribute type defaults to byte. While new attributes may be added to the cds_attributes file, the set of syntaxes (data types) is fixed.
Each entity has a `show` command that displays the names and values of the specified attributes. When you display an attribute that has more than one value, the `show` command lists the attribute name followed by each value of the attribute. For more information about CDS attributes, see the DCE Directory Service information in the *z/OS DCE Administration Guide*.

**Time Offset:** When the DTS entity is running on the system where the entry is created, CDS uses the shared memory time (software time). When the DTS entity is not running, however, CDS uses the hardware clock. In both cases, an offset time (depending on the local time zone) is added. OMVS DTS uses a software clock.

**Entering CDS Internal Commands:** CDS internal commands can be abbreviated as well as continued beyond one line. To abbreviate a command, you can enter either its first four characters or its unique abbreviated name. The unique abbreviated name for each command is listed in the **Format** section of the command description; characters that do not have to be entered are enclosed in square brackets. For example, the unique abbreviated name for the `show directory` command is `sh d`. As a result, the following two commands are equivalent:

```
cdscp> show directory /./sales

cdscp> sh d /./sales
```

CDS internal commands may also be continued beyond one line. The back slash (\) and the dash (-) can be used as continuation characters for `cdscp` commands. Therefore, both of the following are valid:

```
cdscp> show \n
    _> dir /.//*

    and

    cdscp> show -

    _> cdscp -

    _> confidence
```

When the last characters of the first command line do not form a complete word, enter a back slash (\) or dash and continue to the next line. However, when the last characters of the first command line do form a complete word, precede the back slash or dash with a space as illustrated below:

```
cdscp> set link /./sales CDS_LinkTimeout \n
    _> = (1996-12-31-12:00:00 090-00:00:00)
```

While the back slash continues a command beyond one line, the forward slash (/) separates elements in a full name. Multiple consecutive forward slashes (///) are compressed into a single forward slash (/). The forward slash is not permitted in a simple name.

**Using Wildcard Characters:** When entering a name in `show` and `list` commands, you can use wildcard characters in the rightmost simple name (the name to the right of the last slash (/) in the full path name). The asterisk (*) matches 0 or more characters in a simple name. The question mark (?) matches exactly one character in a simple name.

When you use an asterisk or a question mark as a usual character in the rightmost simple name of a `show` or `list` command, precede it with a back slash escape character (\* or \?). Otherwise, the character is interpreted as a wildcard.

You cannot use wildcard characters in the `show clerk` command.
Using Comment Characters: Within a file in the shell, both the # and ! characters are comment starters for all cdscp commands. Any string following these characters are ignored (including the comment character itself). The character cannot be preceded by the back slash escape character. The only way to include it in your entry name or attribute value is to enclose the simple name or attribute value in a pair of single or double quotation marks, respectively.

Note: Comment characters are only applicable in a file within a shell, not when entering cdscp commands.

Privilege Required: CDS supports the following DCE permissions: read (r), write (w), insert (i), delete (d), test (t), control (c), and administrator (a). Each permission has a slightly different meaning, depending on the kind of CDS name with which it is associated. In general, the permissions are defined as follows:

- **read** permission allows a principal to look up a name and view the attribute values associated with it.
- **write** permission allows a principal to change the modifiable attributes associated with a name, except the name’s access control list (ACL) entries.
- **insert** permission (for use with directory entries only) allows a principal to create new names in a directory.
- **delete** permission allows a principal to delete a name from the name space.
- **test** permission allows a principal to test whether an attribute of a name has a particular value without being able to actually see any of the values (that is, without having read permission to the name). Test permission provides application programs with a more efficient way to verify a CDS attribute value. Rather than reading an entire set of values, an application can test for the presence of a particular value.
- **control** permission allows a principal to modify the ACL entries associated with a name, and is automatically granted to the creator of a CDS name.

  Note: Read permission is also necessary for modifying the ACLs of a CDS entry; otherwise, acl_edit is not able to bind to the entry.

- **admin** permission (for use with directory entries only) allows a principal to enter CDS control program commands that control the replication of directories.

The creator of a name is automatically granted all permissions appropriate for the type of name created. For example, a principal creating an object entry is granted read, write, delete, test, and control permission to the object entry. A principal creating a directory is granted read, write, insert, delete, test, control, and administrator permission to the directory.

Examples

The following command starts the CDS control program:

```
READY
cdscp
cdscp>
```

The file $HOME/.cdscpinit is the startup file. It can contain cdscp commands that run each time cdscp is started. For example, you can set your cdscp confidence level to high and set your cdscp preferred clearinghouse when you start cdscp by putting these commands in the $HOME/.cdscpinit file.

The following command operates from the system prompt to display the attributes of the CDS clerk on the local system:
cdscp

READY
   cdscp show clerk

The following is a shell example:

#!/bin/sh
# show attributes for three directories
dirlist="dir1 dir2 dir3"
for dir in $dirlist
do
   cdscp show dir /.:/$dir
done

530 Command Reference
clear cached server

Removes knowledge of a specifically defined server from the local clerk's cache.

Format

cdscp cl[ear] ca[ched] s[erver] name

Parameters

name The simple name given to the cached server when it is created.

Usage

The clear cached server command removes knowledge of a server from the local clerk's cache. You can only clear servers that you have specifically created with the define cached server command.

Privilege Required: You must have w (write) permission to the clerk.

Note: The clear cached server command is supported in z/OS DCE by the dcecp cdscache delete command.

Examples

The following command removes knowledge of the server nrl from the clerk cache:

cdscp> clear cached server nrl

Related Information

Commands:

define cached server     dump clerk cache     show cached server

Books:

- z/OS DCE Administration Guide
clear clearinghouse

Removes knowledge of a clearinghouse from the server's memory.

Format

cdscp cl[ear] cl[earinghouse] clearinghouse-name

Parameters

clearinghouse-name
The full name of the clearinghouse.

Usage

The clear clearinghouse command removes knowledge of the specified clearinghouse from the server's memory. The clearinghouse files are not deleted. This ensures that the clearinghouse is not automatically enabled on server restarts. If you enter a list clearinghouse command, the clearinghouse will still be listed.

Before you can delete a cleared clearinghouse, you must use the create clearinghouse command to recreate it. After recreating the clearinghouse, you can use the delete clearinghouse command to remove it.

This command is part of the process of relocating a clearinghouse. See the z/OS DCE Administration Guide for more information.

Privilege Required: You must have w (write) permission to the server on which the clearinghouse resides.

Examples

The following command clears the clearinghouse /.:/Paris2_CH to prepare for moving it to another server:

cdscp clear clearinghouse /.:/Paris2_CH

Related Information

Commands:
create clearinghouse
delete clearinghouse
list clearinghouse
set cdscp preferred
clearinghouse
show cdscp preferred
clearinghouse
show clearinghouse
create child

Creates a child pointer in the master replica of the parent directory.

Format

cdscp cr[eate] ch[ild] child-name cl[earinghouse] clearinghouse-name

Parameters

child-name The full name of the child pointer.

clearinghouse-name The full name of a clearinghouse that contains a replica of the child directory.

Usage

A child pointer connects a parent and child directory. The child pointer is stored in the parent directory and has the same name as the child directory.

The create child command creates a child pointer in the master replica of the parent directory. When CDS looks up a name in the name space, it uses child pointers to locate directory replicas. Use the set cdscp preferred clearinghouse command before issuing this command to ensure that the request is directed to the master replica.

The clearinghouse must be inside the cell.

Note: The parent directory is implied by the child-name value.

Privilege Required: You must have i (insert) permission to the parent directory.

Note: Use the create child command only to re-create a child pointer that is accidentally deleted. This command is designed only for diagnostics work. This command will fail if the associated directory does not exist. If the associated directory exists, this command will return successfully.

Examples

The following command creates the child pointer in the parent directory ./subsys. It uses the replica located at the ./NY_CH clearinghouse to fill in its replica set.

cdscp> create child ./subsys/childptr clearinghouse ./NY_CH

Related Information

Commands:

delete child               list child             show child

Books:

• z/OS DCE Administration Guide
create clearinghouse

Creates a clearinghouse or makes an existing clearinghouse available.

Format

cdscp cr[eate] cl[earinghouse] clearinghouse-name

Parameters

clearinghouse-name

The full name of the clearinghouse.

Usage

The create clearinghouse command creates a clearinghouse on the local server system or makes an existing clearinghouse available. The server startup command usually creates a new clearinghouse when you configure a new Cell Directory Service (CDS) server. You may need to create a second clearinghouse on a particular server; for example, if you are temporarily relocating a clearinghouse on a different server. See the z/OS DCE Administration Guide for more information about relocating a clearinghouse.

Clearinghouses should be named only in the root. When you enter the create clearinghouse command, CDS creates a read-only replica of the root directory and stores it in the new clearinghouse as the initial replica. Because the process that creates the new clearinghouse start a skulk of the root directory, all replicas of the root should be reachable when you enter the command.

This command is usually processed only by the network configuration procedure. To ensure that all replicas of the root are reachable, perform an immediate skulk of .:/ prior to entering this command.

Privilege Required: You need w (write) permission to the server on that you intend to create the clearinghouse and A (Admin) permission to the cell root directory. The server principal needs r (read), w (write), and A (Admin) permissions to the cell root directory.

Examples

The following command creates a clearinghouse named .:/Boston_CH on the local server system:

```
cdscp> create clearinghouse .:/Boston_CH
```

Related Information

Commands:

- clear clearinghouse
- delete clearinghouse
- list clearinghouse
- set cdscp preferred clearinghouse
- show cached clearinghouse
- show cdscp preferred clearinghouse
- show clearinghouse
create directory

Creates a directory.

Format

cdscp cr\{create\} d\{irectory\} directory-name [c\{learinghouse\} clearinghouse-name

Parameters

directory-name The full name of the directory.
clearinghouse-name The name of the clearinghouse in which you want to create the directory.

Usage

The create directory command creates a directory with the name that you specify. If you do not specify a clearinghouse, CDS creates the master replica of the directory in the same clearinghouse as the new directory's parent directory, implied in the fully qualified directory name. The clearinghouse must be inside the cell.

Privilege Required: You must have r (read) and i (insert) permission to the parent directory, and w (write) permission to the clearinghouse that stores the master replica of the new directory. The server principal must have r (read) and i (insert) permission to the parent directory of the new directory.

Notes:
1. To ensure that all replicas are consistent, perform an immediate skulk of the parent directory after issuing this command.
2. The create directory command is supported in z/OS DCE by the dcecp directory create command.

Examples

The following command creates a directory named ./sales:

cdscp> create directory ./sales

Related Information

Commands:

add directory delete directory remove directory show directory
create replica list directory set directory

Books:

• z/OS DCE Administration Guide
create link

Creates a soft link and optionally specifies an expiration time and an extension time.

Format

cdscp create [link] link-name CDS_LinkTarget [CDS_LinkTimeout] [target-name] [CDS_LinkTimeout] [CDS_LinkTimeout]

(expiration-time extension-time)

Parameters

- link-name: The full name of the soft link.
- target-name: The full name of the entry to which the soft link points.
- expiration-time: A date and time after which CDS checks for existence of the soft link's target and either extends or deletes the soft link. The value is specified as yyyy-mm-dd-hh:mm:ss (year-month-day-hour:minute:second). You can abbreviate this value.
- extension-time: A period of time by which to extend the soft link's expiration time (if the server has validated that the target still exists). The value is specified as ddd-hh:mm:ss (days-hour:minute:second). You can abbreviate this value.

There are no default values for expiration-time and extension-time.

Usage

The create link command creates a soft link. If you specify the CDS_LinkTimeout attribute, you must specify an expiration time and an extension time. If you omit the CDS_LinkTimeout attribute, the soft link is permanent and must be explicitly deleted.

Privilege Required: You must have I (insert) permission to the directory where you are creating the soft link.

Note: The create link command is supported in z/OS DCE by the dcecp link create command.

Examples

The following command creates a soft link, relative to the expiration time, named ./sales/tokyo/price_server that points to an object entry named ./sales/east/price_server. The expiration value specifies that CDS will check that the destination name ./sales/east/price_server still exists on June 25, 1997, at 12:00 p.m. If the destination name still exists, the soft link remains in effect another 90 days.

cdscp> create link ./sales/tokyo/price_server CDS_LinkTarget \
> = ./sales/east/price_server CDS_LinkTimeout = (1997-06-25-12:00:00 \
> 90-00:00:00)
Related Information

Commands:

- delete link
- list link
- remove link
- set link
- show link

Books:

- [z/OS DCE Administration Guide](#)
create object

Creates an object entry.

Format

```
cdscp create object object-name [CDS_Class [=] class-name CDS_ClassVersion [=] value]
```

Parameters

- **object-name**: The full name of the object entry.
- **class-name**: The class of object entry being created. You can specify an application-defined class name. A class is specified as a simple name limited to 31 characters.
- **value**: The version of the class assigned to the object entry. Specify the value as v.n, where v defines the major release number and n specifies the minor version number. The value of v and n can be a positive integer from 0–255. Specifying a class version is useful for allowing the definition of a class to evolve as the application is revised.

There are no default values for class-name and value.

Usage

The `create object` command creates an object entry. This task is usually done through a client application.

**Note**: For information on how DCE applications use the CDS_Class and CDS_ClassVersion attributes using X/Open Data Services (XDS), see the section on creating new CDS attributes in the chapter describing XDS and the DCE cell name space in the [z/OS DCE Application Development Guide: Core Components](https://www.ibm.com/docs/en/zos-dce?topic=core-components).

**Privilege Required**: You must have i (insert) permission to the parent directory.

**Note**: The `create object` command is supported in z/OS DCE by the `dcecp object create` command.

Examples

The following command creates an object entry named `./sales/east/floor1cp`.

```
cdscp> create object ./sales/east/floor1cp CDS_Class = printer CDS_ClassVersion = 1.0
```

Related Information

**Commands**:

- add object
- delete object
- list object
- remove object
- set object
- show object

**Books**:

create replica

Creates a replica of an existing directory in the specified clearinghouse.

Format

cdscp cr[eate] r[eplica] directory-name c[learinghouse] clearinghouse-name

Parameters

directory-name  The full name of the directory.

clearinghouse-name  The full name of the clearinghouse in which you want to create the replica.

Usage

The create replica command creates a replica of an existing directory in the specified clearinghouse. The clearinghouse must be inside the cell.

Privilege Required:  You must have A (Admin) permission to the directory you intend to replicate and w (write) permission to the clearinghouse that stores the new replica. The server principal must have r (read), w (write), and A (Admin) permission to the directory you intend to replicate.

Notes:

1. This command does not replicate subtrees. To ensure that all replicas are consistent, perform an immediate skulk of the parent directory after issuing this command.

2. The create replica command is supported in z/OS DCE by the dcecp directory create command.

Examples

The following command creates a replica of the .:/mfg directory in the clearinghouse .:/Paris1_CH.


cdscp> create replica .:/mfg clearinghouse .:/Paris1_CH

Related Information

Commands:

create directory  set directory to new epoch  show replica
delete replica  set directory to skulk

Books:

- z/OS DCE Administration Guide
define cached server

Creates knowledge of a server in the local clerk's cache.

Format

cdscp define [cached] server name tower value

Parameters

name A simple name for the cached server.

value The protocol sequence and network address of the server node. The format is protocol-sequence:network-address. A protocol-sequence is a character string identifying the network protocols used to establish a relationship between a client and server. There are two choices of protocol sequences, depending on the network address that is supplied in the binding; ncadg_ip_udp and ncacn_ip_tcp. For the network-address, specify an Internet address using the common Internet address notation. For more information about this format, see the RPC introduction in the z/OS DCE Application Development Reference.

Usage

The define cached server command creates knowledge of a server in the local clerk's cache. This command is typically used to manually provide configuration information to a clerk that cannot automatically configure itself. The clerk must have addressing information for at least one server. When the clerk knows about one server, it can find other servers through referrals. This command is also used for providing the clerk with addressing information about a server that is located outside the local area network (LAN) which cannot be acquired through advertisements, or when AnyNet® support is used.

Note:

In z/OS DCE, you cannot use the same server name for more than one network address, and cannot use the same network address for more than one server name.

Privilege Required: You must have w (write) permission to the clerk.

Note: The define cached server command is supported in z/OS DCE by the dcecp cdscache create command.

Examples

The following command creates knowledge of the server nrl in the local clerk's cache:

cdscp> define cached server nrl tower ncadg_ip_udp:16.20.15.25

Note: The address used must be a valid address in the cell.
Related Information

Commands:

- clear cached server
- show cached server
- dump clerk cache

Books:

- Z/OS DCE Administration Guide
delete child

Deletes a child pointer from the name space.

Format

cdscp del[ete] ch[ild] child-name

Parameters

child-name The full name of the child pointer.

Usage

The delete child command deletes a child pointer from the name space. Use this command for debugging purposes only.

Privilege Required: You must have d (delete) permission to the child pointer or administrator permission to the parent directory.

Note: Use the delete child command only when the directory to which the child pointer refers is deleted, and the child pointer accidentally remains.

Examples

The following command deletes the child pointer that accidentally remains after the /.:/sales/east directory is deleted:

cdscp> delete child /.:/sales/east

Related Information

Commands:

create child list child show child

Books:

- z/OS DCE Administration Guide
delete clearinghouse

Deletes the specified clearinghouse from the local server system.

Format

cdscp del[ete] cl[earinghouse] clearinghouse-name

Parameters

clearinghouse-name
    The full name of the clearinghouse.

Usage

The delete clearinghouse command deletes a clearinghouse from the local server system. The CDS does not permit you to delete a cleared clearinghouse. Before you can delete a cleared clearinghouse, you must recreate it using the create clearinghouse command.

The delete clearinghouse command automatically deletes all read-only replicas from a clearinghouse. CDS does not permit you to delete a clearinghouse that contains a master replica. See the z/OS DCE Administration Guide for more information about handling master replicas when deleting a clearinghouse.

It is recommended that you delete all replicas except the root before issuing this command.

Privilege Required: You must have w (write) and d (delete) permission to the clearinghouse and A (Admin) permission to all directories that store replicas in the clearinghouse. The server principal needs d (delete) permission to the associated clearinghouse object entry and A (Admin) permission to all directories that store replicas in the clearinghouse.

Examples

The following command deletes a clearinghouse named /.:/sales/Orion_CH from the local server system:

cdscp delete clearinghouse /.:/sales/Orion_CH

Related Information

Commands:

clear clearinghouse
create clearinghouse
list clearinghouse
set cdscp preferred clearinghouse
show clearinghouse
show cdscp_preferred clearinghouse
delete directory

Deletes a directory.

Format

cdscp del[ete] d[irectory] directory-name

Parameters

directory-name The full name of the directory.

Usage

The delete directory command deletes a directory. The directory cannot contain any object entries, soft links, or child pointers. The master replica of the directory must be the only remaining replica in the cell. Use the delete replica command if you have to remove read-only replicas. You can use:

\[\text{dcecp -c directory delete } /.:/name -tree\]

to remove directory contents and children.

Privilege Required: You must have d (delete) permission to the directory and w (write) permission to the clearinghouse that stores the master replica of the directory. The server principal must have A (Admin) permission to the parent directory or d (delete) permission to the child pointer that points to the directory you want to delete.

Note: The delete directory command is supported in z/OS DCE by the dcecp directory delete command.

Examples

The following command deletes the directory /.:/eng from the name space:

\[\text{cdscp> delete directory } /.:/eng\]

Related Information

Commands:

- add directory
- create directory
- delete replica
- list directory
- remove directory
- set directory
- set directory to skulk
- show directory

Books:
- z/OS DCE Administration Guide
delete link

Deletes a soft link.

Format

cdscp del[ete] l[ink] link-name

Parameters

link-name — The full name of the soft link.

Usage

The delete link command deletes a soft link.

Privilege Required: You must have d (delete) permission to the soft link, or A (Admin) permission to the directory that stores the soft link.

Note: The delete link command is supported in z/OS DCE by the dcecp link delete command.

Examples

The following command deletes the soft link /./sales/asia:

cdscp> delete link /./sales/asia

Related Information

Commands:

create link remove link show link
list link set link

Books:

- z/OS DCE Administration Guide
delete object

Deletes an object entry.

Format

cdscp del[ete] [object] object-name

Parameters

object-name The full name of the object entry.

Usage

The delete object command deletes an object entry. This task is usually done through the client application, except under certain circumstances (for example, if the application is obsolete or no longer has access to the name space). All attributes stored in the object are deleted.

Privilege Required: You must have d (delete) permission to the object entry, or A (Admin) permission to the directory that stores the object entry.

Note: The delete object command is supported in z/OS DCE by the dcecp object delete command.

Examples

The delete object command can be used to delete the object entry ./sales/east/floor1pr2 in the following manner:

cdscp> delete object ./sales/east/floor1pr2

Related Information

Commands:

add object list object set object
create object remove object show object

Books:

- z/OS DCE Administration Guide
delete replica

Deletes a read-only replica of a directory from a clearinghouse.

Format

cdscp del[ete] r[eplica] directory-name cl[earinghouse] clearinghouse-name

Parameters

directory-name The full name of the directory.
clearinghouse-name The full name of the clearinghouse where the replica resides.

Usage

The delete replica command deletes a read-only replica of a directory from a clearinghouse. Use the delete directory command to delete the master replica of the directory, only after all other replicas of the directory are deleted. The clearinghouse must be inside the cell.

Privilege Required: You must have A (Admin) permission to the directory whose replica you want to delete and w (write) permission to the clearinghouse from which you are deleting the replica.

Note: The delete replica command is supported in z/OS DCE by the dcecp directory delete command.

Examples

The following command deletes a read-only replica of the ./mfg directory from the ./Paris1_CH clearinghouse:

cdscp> delete replica ./mfg clearinghouse ./Paris1_CH

Related Information

Commands:
create replica delete directory show replica

Books:
• Z/OS DCE Administration Guide
disable clerk

Stops the clerk on the local system.

Format

cdscp dis[able] c[lerk]

Usage

The disable clerk command stops the clerk on the local system. All active communication with any server is stopped, all client calls in progress fail, and the advertiser is shut down. The clerk cache is copied to the disk.

To bring the advertiser and clerk back up, use the modify DCEKERN command to bring up the advertiser first, then use the modify DCEKERN command again to bring up the clerk.

Privilege Required: You must have w (write) permission to the clerk.

Note: If you are disabling a clerk on a system where a server is running, make sure you disable the server first.

Examples

The following command stops the clerk on the local server system:

cdscp> disable clerk

Related Information

Commands:

show clerk

Books:

- z/OS DCE Administration Guide
- z/OS DCE Configuring and Getting Started
disable server

Stops the server on the local system.

Format

cdscp disable server

Usage

The `disable server` command stops the server on the local system. The server is disabled after all transactions in progress are completed.

Privilege Required: You must have `w (write)` permission to the server.

Examples

The following command stops the server on the local system:

cdscp disable server

Related Information

Command:
show server
do

Runs cdscp commands contained in a specified file.

Format

cdscp do filename

Parameters

filename The HFS file name that contains a list of CDSCP commands to be run.

Usage

The do command permits you to run a list of CDSCP commands without having to manually enter them individually. This command is useful for executing a specific set of commands repeatedly in interactive mode. To add a comment in this file, use the # or ! character. All characters on the same line following the # or ! character are ignored with the following exceptions:

- If the character is part of a simple name (of a CDS entry name), enclosed by a pair of either single or double quotation marks.
- If the character is part of an attribute value enclosed by a pair of either single or double quotation marks.

Quotation marks are removed from single and double quoted strings. The back slash escape character (\) cannot be used with these characters. The startup file for cdscp is $HOME/.cdscpinit.

Privilege Required: You must have r (read) permission to the file in HFS that is run each time cdscp is started.

Examples

If file /u/user1/cdscp.dofile contains the following lines,

create directory /.../dir1
list directory /.../dir1
delete directory /.../dir1

then running the do file displays

```
cdscp> do /u/user1/cdscp.dofile
  LIST
  DIRECTORY /.../abc.com/dir1

dir1
```

dscp>

There is no indication that the create and delete commands were processed, because these commands do not return any message if the operation was successful. Only error messages and the show, list, and dump CDSCP commands return information.
dump clerk cache

Displays the contents of the clerk cache.

Format

cdscp du[mp] c[lerk] c[ache]

Usage

The dump clerk cache command displays the contents of the clerk cache on the screen. Use this command when you are solving CDS problems.

Privilege Required: You must have superuser or root uid=0 authority on the clerk system. No CDS permissions are required.

Note: The dump clerk cache command is supported in z/OS DCE by the dcecp cdscache dump command.

Examples

The following command displays the contents of the clerk cache on the screen:

cdscp> dump clerk cache

Related Information

Commands:

show clerk

Books:

- *z/OS DCE Administration Guide*
help

Displays a list of available cdscp commands.

Format

cdscp help

Usage

Although the help command displays a list of available cdscp commands, specific information regarding the syntax and arguments of each is not displayed. For detailed syntax, use the cdscp show cdscp syntax command.

Examples

The following example shows the list of available cdscp commands:
usage: cdscp operation [options] [args]

add directory
add object
clear cached server
clear clearinghouse
create child
create clearinghouse
create directory
create link
create object
create replica
define cached server
delete child
delete clearinghouse
delete directory
delete link
delete object
delete replica
disable clerk
disable server
do filename
dump clerk cache
help
list child
list clearinghouse
list directory
list link
list object
remove cellname
remove directory
remove link
remove object
set cdscp confidence
set cdscp preferred clearinghouse
set directory
set directory to new epoch
set directory to skulk
set link
set object
show cached clearinghouse
show cached server
show cdscp confidence
show cdscp preferred clearinghouse
show cdscp syntax
show cell
show child
show clearinghouse
show clerk
show directory
show link
show object
show replica
show server

Chapter 4. Cell Directory Service  553
Related Information

Commands:

show cdp scp syntax
list child

Displays a list of all the child pointers whose names match the specified child name.

Format


Parameters

child-name The full name of a specific child pointer. The last simple name can contain wildcard characters.

attribute-name The name of a particular attribute. For a list of valid attribute names, refer to "show child" on page 590.

attribute-value The value of a particular attribute. Information about the formats required for entering values of various data types is found in "Attributes" on page 525.

Usage

The list child command displays a list of all the child pointers whose names match the specified child name. The last simple name can contain wildcard characters. You can use a with attribute-name [=] attribute-value clause to limit output only to child pointers whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=).

Privilege Required: You must have r (read) permission to the directory that stores the child pointer. If you use a with attribute-name [=] attribute-value clause in the command, you also must have r (read) or t (test) permission to the selected child pointers.

Note: The list child command is supported in z/OS DCE by the dcecp directory list command.

Examples

The following command displays a list of all the child pointers named in the ./sales directory:

cdscp> list child ./sales/*

LIST
CHILD  /.../abc.com/sales
AT  1996-10-15-15:56:00

Related Information

Commands:
create child delete child show child

Books:
- z/OS DCE Administration Guide
list clearinghouse

Displays a list of all the clearinghouses whose names match the specified clearinghouse name.

Format


Parameters

clearinghouse-name
The full name of a specific clearinghouse. The last simple name can contain wildcard characters.

attribute-name
The name of a particular attribute. For a list of valid attribute names, refer to "show clearinghouse" on page 592.

attribute-value
The value of a particular attribute. Information about the formats required for entering values of various data types is found in "Attributes" on page 525.

Usage

The list clearinghouse command displays a list of all the clearinghouses whose names match the specified name. The last simple name can contain wildcards. You can use a with attribute-name [=] attribute-value clause to limit output only to clearinghouses whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=).

Privilege Required: You must have r (read) permission to the directory that stores the associated clearinghouse object entry. If you use a with attribute-name [=] attribute-value clause in the command, you also must have r (read) or t (test) permission to the selected clearinghouses.

Note: The list clearinghouse command is supported in z/OS DCE by the dcecp clearinghouse catalog command.

Examples

The following command displays a list of all the clearinghouses named in the root directory:

cdscp> list clearinghouse /.*/*

    LIST
    CLEARINGHOUSE /.../abc.com/*
    AT 1996-10-15-15:56:00
Munich_CH
Paris_CH
Related Information

Commands:

- show cached clearinghouse
- set cdscp preferred clearinghouse
- show cdscp preferred clearinghouse
- show clearinghouse

Books:

- `z/OS DCE Administration Guide`
list directory

Displays a list of all the directories whose names match the specified directory name.

Format


Parameters

directory-name The full name of a specific directory. The last simple name can contain wildcard characters.

attribute-name The name of a particular attribute. For a list of valid attribute names, refer to "show directory" on page 597. An application-defined attribute is also valid here. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes that your application uses.

attribute-value The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in "Attributes" on page 525.

Usage

The list directory command displays a list of all the directories whose names match the specified directory name. The last simple name can contain wildcards. You can use a with attribute-name [=] attribute-value clause to limit output to directories whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=).

Privilege Required: You must have r (read) permission to the parent directory. If you use a with attribute-name [=] attribute-value clause in the command, you also must have r (read) or t (test) permission to the selected directories.

Note: The list directory command is supported in z/OS DCE by the dcecp directory list command.

Examples

The following command displays the names of all the directories in the /.:/sales directory:

cdscp> list directory /.:/sales/*

LIST
DIRECTORY /.../abc.com/sales

atlanta
austin
boston
chicago
ontario
ny
seattle
Related Information

Commands:

- add directory
- create directory
- delete directory
- remove directory
- set directory
- set directory to skulk
- show directory

Books:

- z/OS DCE Administration Guide
list link

Displays a list of all the soft links whose names match the link name that you specify.

Format


Parameters

link-name The full name of a specific soft link. The last simple name can contain wildcard characters.

attribute-name The name of a particular attribute. For a list of valid attribute names, refer to "show link" on page 600.

attribute-value The value of a particular attribute. Information about the formats required for entering values of various data types is found in "Attributes" on page 525.

Usage

The list link command displays a list of all the soft links whose names match the link name that you specify. The last simple name can contain wildcard characters. You can use a with attribute-name [=] attribute-value clause to limit output only to soft links whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=). This command does not list the name of the directory, object entry, or other soft link to which the soft link points.

Privilege Required: You must have r (read) permission to the directory that stores the soft link. If you use a with attribute-name [=] attribute-value clause in the command, you also must have r (read) or t (test) permission to the selected soft links.

Note: The list link command is supported in z/OS DCE by the dcecp link show command.

Examples

The following command displays a list of all the soft links whose names begin with the letter l in the directory ./admin:

cdscp> list link ././admin/l*

LIST
SOFTLINK /.../abc.com/admin
lnk01
lnk02
lnk03
lnk04
lnk05
lnk06

Related Information

Commands:
create link  remove link  show link
delete link  set link

Books:

- z/OS DCE Administration Guide
list object

Lists the specific object entries (including clearinghouse object entries) whose names match the object entry name specified.

Format


Parameters

object-name  The full name of a specific object entry. The last simple name can contain wildcard characters.

attribute-name  The name of a particular attribute. For a list of valid attribute names, refer to “show object” on page 602. An application-defined attribute is also valid here. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes that your application uses.

attribute-value  The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in “Attributes” on page 525.

Usage

The list object command displays a list of all the object entries (including clearinghouse object entries) whose names match the object entry name that you specify. The last simple name can contain wildcard characters. You can use a with attribute-name [=] attribute-value clause to limit output to only those object entries whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=).

Privilege Required:  You must have r (read) permission to the directory that stores the object entry. If you use a with attribute-name [=] attribute-value clause in the command, you also must have r (read) or t (test) permission to the selected object entries.

Note:  The list object command is supported in z/OS DCE by the dcecp object show command.

Examples

The following command displays a list of all the object entries named in the directory .:/eng:

cdscp> list object .:/eng/*

LIST
OBJECT  /.../abc.com/eng

juno
test_stats
work_disk1
work_disk2
Related Information

Commands:

- add object
- create object
- delete object
- remove object
- set object
- show object

Books:

- z/OS DCE Administration Guide
remove directory

Removes a value from a set-valued or single-valued attribute (including application-defined attributes) of a directory.

Format

```bash
cdscp r[emove] d[irectory] directory-name attribute-name [ [=} attribute-value]
```

Parameters

directory-name  The full name of the directory.

attribute-name  The name of a particular attribute. Specify only one attribute at a time. See the `cds_attributes` file (`/opt/dcelocal/etc/cds_attributes`) for the list of attributes and corresponding data types that your application uses.

attribute-value  The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the `cds_attributes` file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in "Attributes" on page 525.

Usage

The `remove directory` command deletes attributes created by the `add directory` and `set directory` commands. Usually this task is performed through the client application. If no attribute value is specified, the command deletes the entire attribute. When you are using the `remove directory` command with the attribute value option, a space must precede and follow the equal sign (=).

The attribute value option can only be used on a set-valued attribute. An error message will be returned if an attribute value is specified on the `remove directory` command with a single-valued attribute name specified. See the Directory Service section of the `z/OS DCE Administration Guide` for more information about attributes.

When you remove all values of a multi-valued attribute, the attribute name displays on the `show directory` command, with the value:

```
<empty set>
```

Privilege Required:  You must have w (write) permission to the directory.

Note:  The `remove directory` command is supported in z/OS DCE by the `dcecp directory modify` command.

Examples

To remove the value 5 from the user-defined, set-valued attribute dirregion of a directory named `./sales`, follow these steps:

1. Read the `cds_attributes` file to check that the attribute dirregion is listed, as shown in the following display:

```
<table>
<thead>
<tr>
<th>OID</th>
<th>LABEL</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.22.1.3.66</td>
<td>dirregion</td>
<td>small</td>
</tr>
</tbody>
</table>
```

2. Enter the following command to remove the value 5 from the attribute dirregion.
cdscp> remove directory ./:/sales dirregion = 5

Related Information

Commands:

- add directory
- create directory
- delete directory
- list directory
- set directory
- set directory to skulk
- show directory

Books:

- z/OS DCE Administration Guide
remove link

Removes the timeout value attribute of a soft link.

Format

cdscp r[emove] l[ink] link-name CDS_LinkTimeout

Parameters

link-name The full name of the soft link.

Usage

The remove link command removes a timeout value attribute of a soft link, CDS_LinkTimeout, causing the soft link to become permanent.

The CDS_LinkTimeout is the only removable attribute of a link.

Privilege Required: You must have w (write) permission to the soft link.

Note: The remove link command is supported in z/OS DCE by the dcecp link modify command.

Examples

The following command removes the timeout value attribute of a soft link named ./:/eng/link01:

cdscp> remove link ./:/eng/link01 CDS_LinkTimeout

Related Information

Commands:
create link          list link          show link
delete link          set link

Books:
• z/OS DCE Administration Guide
remove object

Removes a value from a set-valued or single-valued attribute (including application-defined attributes) of an object entry.

Format

cdscp r[emove] o[bject] object-name attribute-name [ [=] attribute-value]

Parameters

object-name The full name of the object entry.
attribute-name The name of a particular attribute. Specify only one attribute at a time. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes and corresponding data types that your application uses.
attribute-value The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in “Attributes” on page 525.

Usage

The remove object command deletes attributes created by the add object and set object commands. Usually, this task is performed through the client application. If no attribute value is specified, the command will delete the entire attribute. When using the remove object command with the attribute value option, a space must precede and follow the equal sign (=).

The attribute value option can only be used on a set-valued attribute. An error message will be returned if an attribute value is specified on the remove object command with a single-valued attribute name specified. See the Directory Service section of the z/OS DCE Administration Guide for more information about attributes.

Privilege Required: You must have w (write) permission to the object entry.

Note: The remove object command is supported in z/OS DCE by the dcecp object modify command.

Examples

To remove the value ps from the user defined, set-valued, attribute printcap of an object entry named ./mlh/deskprinter, follow these steps:

1. Read the cds_attributes file to check that the attribute printcap is listed, as shown in the following display:

```
  OID     LABEL      SYNTAX
1.3.22.1.3.50 printcap  char
```

2. Enter the following command to remove the value ps from the attribute printcap:

```
cdscp> remove object ./mlh/deskprinter printcap = ps
```
Related Information

Commands:

- add object
- delete object
- create object
- list object
- set object
- show object

Books:

- [z/OS DCE Administration Guide](#)
set cdscp confidence

Sets the confidence level of CDS control program commands.

Format

cdscp se[t] cd[scp] c[onfidence] [=] value

Parameters

value One of the following confidence levels:

low A low confidence level means that the clerk obtains information from caches or the most convenient server.

medium A medium level means that the clerk obtains information directly from a server.

high A high level means that the clerk obtains information only at master replicas.

The initial value is medium.

Usage

The set cdscp confidence command sets the confidence level (of clerk calls issued as a result) of CDS control program commands. Use this command within the CDS control program. Exiting from the CDS control program removes the confidence level setting. Therefore, each time you enter the control program, you must reset the confidence level. A space must precede and follow the equal sign (=).

Or, you can use the startup file, $HOME/.cdscpinit, to set the confidence level for each time cdscp is started.

Privilege Required: No special privileges are needed to use the set cdscp confidence command.

Examples

The following command sets the confidence level of clerk calls to high:

READY
cdscp
cdscp> set cdscp confidence = high

Related Information

Commands:

show cdscp confidence

Books:

- z/OS DCE Administration Guide
set cdscp preferred clearinghouse

Specifies a preferred clearinghouse to use for satisfying read requests that result from CDS control program commands.

Format

cdscp set[t] cd[scp] p[referred] c[learinghouse] [clearinghouse-name]

Parameters

clearinghouse-name
The full name of the preferred clearinghouse. If you omit this argument, the command causes CDS to revert to the default, which is to use any clearinghouse.

Important Note to Users

When you enter the clearinghouse-name you must make sure that the clearinghouse exists. This command accepts a non-existent or wrong clearinghouse-name, and uses the name of the default clearinghouse. Use the cdscp show preferred clearinghouse command to view the values that have been set.

Usage

The set cdscp preferred clearinghouse command specifies a preferred clearinghouse to use for satisfying read requests that result from CDS control program commands. You cannot specify a preferred clearinghouse for making modifications, because these requests always use the master replica. Use this command within the CDS control program. Exiting from the CDS control program removes the preferred clearinghouse setting. Therefore, each time you enter the control program, you must reset the preferred clearinghouse, or you can add it to the startup file, $HOME/.cdscpinit.

Privilege Required: No special privileges are needed to use the set cdscp preferred clearinghouse command.

Examples

The following command specifies ./Paris_CH as the preferred clearinghouse:

READY
  cdscp
  cdscp> set cdscp preferred clearinghouse ./Paris_CH

Related Information

Commands:

show cdscp preferred clearinghouse

Books:

• z/OS DCE Administration Guide
set directory

Changes the value of a modifiable, single-valued attribute of a directory.

Format

cdscp se[t] d[irectory] directory-name attribute-name [=] attribute-value

Parameters

directory-name The full name of the directory.
attribute-name The name of a particular attribute. Specify only one attribute at a time. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes and corresponding data types that your application uses.
attribute-value The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in “Attributes” on page 525.

Usage

The set directory command changes the value of a modifiable, single-valued attribute of a directory. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. A space must precede and follow the equal sign (=). See the Directory Service section of the z/OS DCE Administration Guide for more information about attributes.

You can specify an application-defined attribute or any of the following attributes:

**CDS_Convergence = value**

Specify one of the following for value:

- **low** CDS does not immediately propagate any updates. The next skulk distributes all updates that occurred since the previous skulk. Skulks occur at least once every 24 hours.
- **medium** CDS attempts to immediately propagate an update to all replicas. If the attempt fails, the software lets the next scheduled skulk make the replicas consistent. Skulks occur at least once every 12 hours.
- **high** CDS attempts to immediately propagate an update to all replicas. If that attempt fails (for example, if one of the replicas is unavailable), a skulk is scheduled for within 1 hour. Background skulks occur at least once every 12 hours. Use this setting temporarily and briefly because it uses extensive system resources.

By default, every directory inherits the convergence setting of its parent at creation time. The default setting on the root directory is medium.

Data_type is Character.

**CDS_InCHName = value**

Indicates whether a directory or any of its descendants can store clearinghouse names. If this value is true

The directory can store clearinghouse names.
set directory

If this value is **false**

The directory cannot store clearinghouse names.

Data_type is Character.

**CDS_UpgradeTo** = v.n

Controls the upgrading of a directory from one version of CDS to another. By modifying this attribute, you can begin the upgrading of a directory to a higher version of CDS. Specify the value as v.n, where v specifies the major version number and n specifies the minor version number. There is no default.

Data_type is Version.

**Privilege Required:** You must have **w** (write) permission to the directory.

**Note:** The set directory command is supported in z/OS DCE by the dcecp directory modify command.

**Examples**

The following command sets a low convergence value on the ./:/mfg directory:

```
cdscp> set directory ./:/mfg CDS_Convergence = low
```

**Related Information**

Commands:

- add directory
- delete directory
- remove directory
- show directory
- create directory
- list directory
- set directory to
- skulk

Books:

- [z/OS DCE Administration Guide](#)
**set directory to new epoch**

Rebuilds a directory replica set, allowing you to designate a new master replica or to exclude a replica.

**Format**

```
```

**Parameters**

- `directory-name`: The full name of the directory.
- `clearinghouse-name`: The full name of the clearinghouse in which an individual replica is located. The first `clearinghouse-name` specifies where the master replica is stored. The clearinghouse must be in the cell.

**Usage**

The `set directory to new epoch` command reconstructs a directory’s replica set, allowing you to designate a new master replica or to exclude a replica. You must list each existing replica and specify whether an existing replica must be included in or excluded from the new replica set. You can include or exclude more than one replica.

When you set a new epoch on a directory, you must disable the clearinghouse containing the replica that is being excluded, usually before the exclude. Use the `disable server` command on the system where the clearinghouse is located. If the server has more than one clearinghouse, all its clearinghouses will be disabled. If you are assigning a new master, both the clearinghouse that contains the current master and the one to contain the new master must be enabled and accessible before you enter this command.

**Privilege Required:** You must have A (Admin) permission to the directory, and the server principal must have A (Admin), r (read), and w (write) permission to the directory. When designating a new master replica, you also must have w (write) permission to the clearinghouse that stores the new master replica, and the server principal must have w (write) permission to the clearinghouse where the replica type is changed to read-only.

**Examples**

The following command sets a new epoch for the directory `/./mfg`. The master replica is in the `/./Paris1_CH` clearinghouse, and the read-only replicas are in the `/./Chicago1_CH` and `/./Seattle_CH` clearinghouses. The new replica set excludes the replica in the `/./NY1_CH` clearinghouse.

```
cdscp> set directory /./mfg to new epoch master /./Paris1_CH \
> readonly /./Chicago1_CH /./Seattle_CH exclude /./NY1_CH
```

**Related Information**

- Commands:
  - `set directory to skulk`
  - `show directory`
  - `show replica`

**Books:**
set directory to new epoch

- z/OS DCE Administration Guide
set directory to skulk

Starts the skulk of a directory immediately.

Format

cdscp se[t] d[irectory] directory-name t[o] s[kulk]

Parameters

directory-name  The full name of the directory.

Usage

The set directory to skulk command starts the skulk of a directory immediately. The CDS control program prompt cdscp> does not return until the skulk is complete. The amount of time for the skulk to complete depends on the location, number, and availability of replicas of the directory.

Privilege Required: You must have A (Admin), w (write), i (insert), or d (delete) permission to the directory. The server principal must have A (Admin), r (read), and w (write) permission to the directory.

Note: The set directory to skulk command is supported in z/OS DCE by the dcecp directory synchronize command.

Examples

The following command starts a skulk on the /./admin directory:

cdscp> set directory /./admin to skulk

Related Information

Commands:

- add directory
- create directory
- delete directory
- list directory
- remove directory
- set directory to new epoch
- show directory

Books:

- z/OS DCE Administration Guide
set link

Changes the value of the modifiable, single-valued attributes of a soft link.

Format

`cdscp set[link] link-name attribute-name [=] attribute-value`

Parameters

- `link-name`: The full name of the soft link.
- `attribute-name`: The name of the attribute to be modified. Specify only one attribute at a time. See the Usage section below for valid attribute names.
- `attribute-value`: The value of a particular attribute. Information about the format required for entering values is found in the Usage section below.

Usage

A space must precede and follow the equal sign (=). In general, this command changes the value of a modifiable, single-valued attribute of a soft link. The following are valid attributes:

- **CDS_LinkTarget** = *fullname*
  Specifies the full name of the directory, object entry, or other soft link to which the soft link points. Data_type = Character.

- **CDS_LinkTimeout** = *(expiration-time extension-time)*
  Specifies a timeout value after which the soft link is checked and possibly deleted. The timeout value contains both an expiration time and an extension time. If a soft link expires and its target entry is deleted, the soft link is deleted. If the soft link still points to an existing entry, its life is extended by the expiration time. Specify `expiration-time` in the format `yyyy-mm-dd-hh:mm:ss` (year-month-day-hour:minute:second). Specify `extension-time` in the format `ddd-hh:mm:ss` (day-hour:minute:second). Data_type = Timeout.

Privilege Required: You must have `w` (write) permission to the soft link.

Note: The `set link` command is supported in z/OS DCE by the `dcecp link modify` command.

Examples

The following command redirects a soft link named `l:/admin/work_disk` from its current destination name, `l:/admin/work_disk01`, to a new destination name, `l:/admin/work_disk03`.

```
cdscp> set link /:/admin/work_disk CDS_LinkTarget = /:/admin/work_disk03
```
Related Information

Commands:

- create link
- delete link
- list link
- remove link
- show link

Books:

- z/OS DCE Administration Guide
set object

Changes the value of a modifiable, single-valued attribute of an object entry.

Format

cdscp se[t] o[bject] object-name attribute-name [=] attribute-value

Parameters

- **object-name**: The full name of the object entry.
- **attribute-name**: The name of the attribute to be modified. Specify only one attribute at a time. See the `cds_attributes` file (`/opt/dce/local/etc/cds_attributes`) for the list of attributes and corresponding data types that your application uses.
- **attribute-value**: The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the `cds_attributes` file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in “Attributes” on page 525.

Usage

The `set object` command changes the value of a modifiable, single-valued attribute of an object entry. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. A space must precede and follow the equal sign (=). See the Directory Service section of the [z/OS DCE Administration Guide](https://www.ibm.com) for more information about attributes.

Privilege Required: You must have `w` (write) permission to the object entry.

Note: The `set object` command is supported in z/OS DCE by the dcecp object modify command.

Examples

The value of the `sales_record` attribute is changed to `region2` of an object entry named `/.:/Q1_records` as follows:

1. Read the `cds_attributes` file to check that the attribute `sales_record` is listed, as shown in the following display:

<table>
<thead>
<tr>
<th>OID</th>
<th>LABEL</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.22.1.3.66</td>
<td>sales_record</td>
<td>char</td>
</tr>
</tbody>
</table>

2. Enter the following command to assign the value `region2` to the attribute `sales_record` of an object entry named `/.:/Q1_records`.

   ```
   cdscp> set object /./Q1_records sales_record = region2
   ```
Related Information

Commands:

- add object
- delete object
- remove object
- create object
- list object
- show object

Books:

- [z/OS DCE Administration Guide](#)
show cached clearinghouse

Display current information about the specified clearinghouse cached in the CDS clerk.

Format

cdscp sh[ow] ca[ched] cl[earinghouse] clearinghouse-name

Parameters

clearinghouse-name
  A specific clearinghouse name. The name can contain wildcard characters.

Usage

The show cached clearinghouse command displays all the names and values of the attributes in the specified cached clearinghouse.

The following data is returned in the command output:

- **Creation Time**: Specifies the time at which this clearinghouse was added to the cache.
- **Miscellaneous Operations**: Specifies the number of operations other than read and write, such as skulks and new epochs, performed by this CDS clerk on the cached clearinghouse.
- **Read Operations**: Specifies the number of lookup operations of any sort performed by the CDS clerk on the cached clearinghouse.
- **Tower**: Specifies the protocol sequence and Internet address of the server that maintains the cached clearinghouse.
- **Write Operations**: Specifies the number of write operations performed by this CDS clerk on the cached clearinghouse.

Privilege Required: You must have r (read) permission to the CDS clerk.

Note: The show cached clearinghouse command is supported in z/OS DCE by the dcecp cdscache show command.

Examples

The following command displays all attributes of the cached clearinghouse /.:/Paris2.CH:

```
cdscp> show cached clearinghouse /.:/Paris2.CH
```

SHOW
Tower = ncadg_ip_udp:14.20.14.32
Creation Time = 1996-10-01-17:03:32.32
Read Operations = 412
Write Operations = 618
Miscellaneous Operations = 278
Related Information

Commands:

list clearinghouse        show clearinghouse

Books:

- [z/OS DCE Administration Guide]
show cached server

Displays address information (tower) of a server in the CDS clerk's cache.

Format

cdscp sh[ow] ca[ched] s[erver] name

Parameters

name A simple name for the cached server. The name can contain wildcard characters.

Usage

The show cached server command displays address information of a server in the CDS clerk's cache.

The following data is returned in the command output:

Name The directory cell name.
Tower The protocol sequence and network address of the server node.

Privilege Required: You must have r (read) permission to the clerk.

Note: The show cached server command is supported in z/OS DCE by the dcecp cdscache show command.

Examples

The following command displays all attributes of the cached server, emv*, where the asterisk (*) matches 0 or more characters in a simple name:

cdscp> show cached server emv*

SHOW CACHED NAMESERVER emv_udp
  Name = /.../emv.abc.com
  Tower = ncadg_ip_udp:14.20.14.32
SHOW CACHED NAMESERVER emv_tcp
  Name = /.../emv.abc.com
  Tower = ncadg_ip_udp:14.20.14.32

Related Information

Commands:
clear cached server define cached server dump clerk cache

Books:
- z/OS DCE Administration Guide
show cdscp confidence

Displays the current confidence level of clerk calls resulting from CDS control program commands.

Format

cdscp sh[ow] cd[scp] c[onfidence]

Usage

The show cdscp confidence command displays the current confidence level of CDSCP commands. A low confidence level means the CDS clerk obtains information from caches or the most convenient server. A medium level means the CDS clerk obtains information directly from a server. A high level means the CDS clerk obtains information only at master replicas.

Use this command from within the CDS control program. Exiting from the CDS control program removes the confidence level setting. The confidence level must be reset each time you enter the CDS control program if you do not want the default setting. Or you can add it to the startup file, $HOME/.cdscpinit.

Privilege Required: No special privileges are needed to use the show cdscp confidence command.

Examples

The following command displays the current confidence level of clerk calls:

READY
cdscp

cdscp> show cdscp confidence

Confidence used is medium

Related Information

Commands:

set cdscp confidence

Books:

• z/OS DCE Administration Guide
show cdscp preferred clearinghouse

show cdscp preferred clearinghouse
Displays the preferred clearinghouse for satisfying read requests that result from CDS control program commands.

Format


Usage

The show cdscp preferred clearinghouse command displays the preferred clearinghouse for satisfying read requests that result from CDS control program commands. You can only read attribute values for entries stored in the specified clearinghouse.

Use this command from within the CDS control program. Exiting from the CDS control program removes the preferred clearinghouse setting. The preferred clearinghouse must be reset each time you enter the CDS control program is entered. Or you can add it to the startup file, $HOME/.cdscpinit.

Privilege Required: No special privileges are needed to use the show cdscp preferred clearinghouse command.

Examples

The following command displays the current clearinghouse:

READY
    cdscp
cdscp> show cdscp preferred clearinghouse

read attribute values from clearinghouse /.../abc.com/Paris_CH

Related Information

Commands:

set cdscp preferred clearinghouse

Books:

- z/OS DCE Administration Guide
show cdscp syntax

Displays the syntax of the available cdscp commands.

**Format**

```plaintext
show cdscp syntax
```

**Usage**

The `help` command displays a list of available cdscp commands, but not specific information regarding the syntax and arguments of each command. Use the `show cdscp syntax` command to display syntax arguments for the available cdscp commands.

**Privilege Required:** No special privileges are needed to use the `show cdscp syntax` command.

**Examples**

The following example shows the list of available cdscp commands with specific syntax and arguments:
show cdscp syntax

cdscp> show cdscp syntax
show cell [cell-name] [as gds|dns]

define cached server name tower protocol-sequence:network-address
show cached server name
clear cached server name

show cached clearinghouse clearinghouse-name

show clearinghouse clearinghouse-name [attribute-name]...
[with attribute-name [=] attribute-value]...
list clearinghouse clearinghouse-name
[with attribute-name [=] attribute-value]...

create directory directory-name [clearinghouse clearinghouse-name]
show directory directory-name [attribute-name]...
[with attribute-name [=] attribute-value]...
list directory directory-name
[with attribute-name [=] attribute-value]...
add directory directory-name attribute-name [=] attribute-value
set directory directory-name attribute-name [=] attribute-value
remove directory directory-name attribute-name [=] attribute-value
delete directory directory-name

set directory directory-name to new epoch master clearinghouse-name
[readonly clearinghouse-name]...
[exclude clearinghouse-name]...
set directory directory-name to skulk

create object object-name [CDS_Class [=] class-name
CDS_ClassVersion [=] value]
show object object-name [attribute-name]...
[with attribute-name [=] attribute-value]...
list object object-name [with attribute-name [=] attribute-value]...
add object object-name attribute-name [=] attribute-value
set object object-name attribute-name [=] attribute-value
remove object object-name attribute-name [=] attribute-value
delete object object-name

create child child-name clearinghouse clearinghouse-name
show child child-name [attribute-name]...
[with attribute-name [=] attribute-value]...
list child child-name [with attribute-name [=] attribute-value]...
delete child child-name

create link link-name CDS_LinkTarget [=] target-name
[CDL_LinkTimeout [=](yyyy-mm-dd-hh:mm:ss ddd-hh:mm:ss)]
show link link-name [attribute-name]...
[with attribute-name [=] attribute-value]...
list link link-name [with attribute-name [=] attribute-value]...
set link link-name attribute-name [=] attribute-value
remove link link-name CDS_LinkTimeout
delete link link-name

create replica directory-name clearinghouse clearinghouse-name
show replica directory-name clearinghouse clearinghouse-name
[attribute-name]
delete replica directory-name clearinghouse clearinghouse-name
show cdscp syntax

dump clerk cache

show clerk
disable clerk

do filename

help

set cdscp confidence [=] high|medium|low
show cdscp confidence

set cdscp preferred clearinghouse [clearinghouse-name]
show cdscp preferred clearinghouse

show cdscp syntax

cdscp>
show cell

Displays the information you require to create a cell entry in either the Domain Name System (DNS) or the Global Directory Service (GDS).

Format

cdscp sh[ow] ce[ll] [cell-name] [a[s] type]

Parameters

cell-name The global name of the cell. If no cell-name is specified, the default cell name is used.
type The global name space in which you want to define the cell. Enter either of the values dns and gds. The default is gds.

Usage

The show cell command displays the information you require to create a cell entry in either the Domain Name System (DNS) or the Global Directory Service (GDS).

Note: DCE does not support cells registered simultaneously in GDS and DNS.

If you want to define a cell in DNS, you can use this command to produce a pre-formatted set of resource records. You can then edit the appropriate DNS data file and copy the output directly into the file. In GDS, cell information is contained in two attributes: CDS_Cell and GDS_Replicas. If you want to define a cell in GDS, you can use this command to obtain the data you require to supply when creating the CDS_Cell and CDS_Replicas attributes. For details, see the Directory Service section in the z/OS DCE Administration Guide.

Privilege Required: You must have r (read) permission to the cell root directory.

Note: The show cell command is supported in z/OS DCE by the dcecp cell show command.

Examples

The following command displays the GDS-formatted output in the local cell:

cdscp> show cell /.../abc.com as gds

SHOW
CELL /.../mycell.goodco.com
Namespace Uuid = 2d2d50ad-8b1a-11ba-8983-08002b0f79aa
Clearinghouse Uuid = 2ab024a8-8b1a-11ba-8983-08002b0f79aa
Clearinghouse Name = /.../abc.com/NY_CH
Replica Type = Master
Tower = ncadg_ip_udp:16.18.17.33
Tower = ncadg_ip_udp:15.18.17.33
Related Information

Books:

- [z/OS DCE Administration Guide](#)
show child

Displays attribute information about the specified child pointer.

Format


Parameters

- \texttt{child-name} The full name of a specific child pointer. The last simple name can contain wildcard characters.
- \texttt{attribute-name} The name of an attribute; see the Attributes section below for valid attribute names.
- \texttt{attribute-value} The value of a particular attribute. Information about the formats required for entering values of various data types is found in "Attributes on page 525".

Usage

The \texttt{show child} command displays the names and values of the attributes specified in \texttt{attribute-name}. You can use a combination of attributes in a single command. Use a space to separate multiple attributes. You can use a \texttt{with attribute-name [=] attribute-value} clause to limit output only to child pointers whose attributes have values equal to the specified values. If you do not supply any attributes, the command displays all attributes and their values. A space must precede and follow the equal sign (=).

Attributes

- \texttt{CDS\_CTS} Specifies the creation timestamp (CTS) of the specified child pointer. Data_type = Timestamp.
- \texttt{CDS\_ObjectUUID} Specifies the unique identifier of the directory to which the child pointer refers. Data_type = UUID.
- \texttt{CDS\_Replicas} Specifies the address, UUID, and name of a set of clearinghouses where a copy of the child directory referenced by the child pointer is located. This attribute also specifies whether the directory in a particular clearinghouse is a master or read-only replica. \texttt{CDS\_Replicas} is not a data type; it is a set of data types like a structure.
- \texttt{CDS\_UTS} Specifies the timestamp of the most recent update to an attribute of the child pointer. Data_type = Timestamp.

Privilege Required: You must have \texttt{r} (read) permission to the child pointer.

Note: The \texttt{show child} command is supported in z/OS DCE by the \texttt{dcecp directory show} command.

Examples

The following command displays all of the attributes and values of the child directory to which the child pointer \texttt{./:/admin} refers:
cdscp> show child ./:/admin

    SHOW
    CHILD  /.../mycell.goodco.com/users
    AT  1996-10-15-15:56:01
    CDS_CTS = 1996-10-15-19:55:52.000000003/08-00-2b-1c-8f-1f
    CDS_UTS = 1996-10-15-19:55:52.000000006/08-00-2b-1c-8f-1f
    CDS_ObjectUUID = 6b5362e8-8b1c-11ca-8981-00002b0f79aa
    CDS_Replicas = :
    Clearinghouse_UUID = 2ab024a8-8b1a-11ca-8981-08002b0f79aa
      Tower = ncadg_ip_tcp:16.18.16.32
      Tower = ncadg_ip_udp:16.18.16.32
    Replica type = master
    Clearinghouse Name = /.../mycell.goodco.com/Boston_CH

**Related Information**

Commands:

- create child
- delete child
- list child

Books:

- [z/OS DCE Administration Guide](#)
show clearinghouse

Displays attribute information about the specified clearinghouse.

Format

\texttt{cdscp sh[ow] c[learinghouse] clearinghouse-name [attribute-name]. . . [w[ith] attribute-name [=] attribute-value]. . .}

Parameters

- \texttt{clearinghouse-name} The full name of a specific clearinghouse. The last simple name can contain wildcard characters.
- \texttt{attribute-name} The name of a particular attribute; see the \texttt{Attributes} section below for valid attribute names.
- \texttt{attribute-value} The value of a particular attribute. Information about the formats required for entering values of various data types is found in \textit{Attributes} on page 525.

Usage

The \texttt{show clearinghouse} command displays the names and values of the attributes specified in \texttt{attribute-name}. You can use a combination of attributes in any sequence in a single command. Use a space to separate multiple attributes. You can use a \texttt{with attribute-name [=] attribute-value} clause to limit output only to clearinghouses whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=). If you do not supply any attributes, the command displays all attributes and their values.

Attributes

- \texttt{CDS\_AllUpTo} Indicates the date and time the clearinghouse object has been updated to reflect the \texttt{CDS\_CHDirectories} attribute. \texttt{Data\_type = Timestamp.}
- \texttt{CDS\_CHDirectories} Specifies the full name and unique identifier (UUID) of every directory that has a replica in this clearinghouse. \texttt{Data\_type = UUID.}
- \texttt{CDS\_CHLastAddress} Specifies the current reported network address of the clearinghouse. \texttt{Data\_type = Tower.}
- \texttt{CDS\_CHName} Specifies the full name of the clearinghouse. \texttt{Data\_type = Character.}
- \texttt{CDS\_CHState} Specifies the state of the clearinghouse. The state \texttt{on} specifies the clearinghouse is running and available:
  1. \texttt{new}
  2. \texttt{on}
  3. \texttt{dying}
\texttt{Data\_type = Character.}
- \texttt{CDS\_CTS} Specifies the creation timestamp (CTS) of the clearinghouse. \texttt{Data\_type = Timestamp.}
CDS_DirectoryVersion
Specifies the current version of the directory in the clearinghouse in which the directory was created. Data_type = Version.

CDS_NSCellname
Specifies the name of the cell in which the clearinghouse resides. Data_type = Character.

CDS_ObjectUUID
Specifies the unique identifier (UUID) of the clearinghouse. Data_type = UUID.

CDS_ReplicaVersion
Specifies the current version of the replica in which the directory was created. Data_type = Version.

CDS_UTS
Specifies the timestamp of the most recent update to an attribute of the clearinghouse. Data_type = Timestamp.

The following internal counters and their values are displayed only when you use this command to display all attributes and their values:

Data Corruption Count
Specifies the number of times that the data corruption event was generated.

Disables
Specifies the number of times that the clearinghouse was disabled since it was last started.

Enables
Specifies the number of times that the clearinghouse was enabled since it was last started.

Read Accesses
Specifies the number of read operations directed to this clearinghouse.

References Returned
Specifies the number of requests directed to this clearinghouse that resulted in the return of a partial answer instead of satisfying the client request.

Skulk Failures
Specifies the number of times that a skulk of a directory, started from this clearinghouse, failed to complete, usually because one of the replicas in the replica set was unreachable.

Entry Missing Count
Specifies the number of times the clearinghouse entry missing event was generated.

Root Not Reachable Count
Specifies the number of times the root lost event was generated.

Upgrades Failed Count
Specifies the number of times that upgrades failed.

Write Accesses
Specifies the number of write operations directed to this clearinghouse.

Privilege Required: You must have r (read) permission to the clearinghouse. If you specify a wildcard clearinghouse name, you also must have r (read) permission to the cell root directory.

Note: The show clearinghouse command is supported in z/OS DCE by the dcecp clearinghouse show command.

Examples
The following command displays the current values of the CDS_UTS and CDS_ObjectUUID attributes associated with the /./Chicago1_CH clearinghouse:
show clearinghouse

cdscp> show clearinghouse /.:/Chicago1_CH CDS_UTS CDS_ObjectUUID

SHOW
CLEARINGHOUSE /.../abc.com/Chicago1_CH
AT 1996-10-21-13:12:30
CDS_UTS = 1996-10-21-13:04:04.000000009/08-00-2b-1c-8f-1f
CDS_ObjectUUID = 3706d70c-8b05-11ca-9002-08002b1c8f1f

Related Information

Commands:
list clearinghouse
set cdscp preferred clearinghouse
show cached clearinghouse
show cdscp preferred clearinghouse

Books:
- z/OS DCE Administration Guide
show clerk

Displays attribute information about the CDS clerk on the local system.

Format

cdscp sh[ow] cler[k]

Usage

The show clerk command displays all the names and values of the CDS clerk attributes on the local system. The CDS clerk must be enabled when you use this command.

The following data is returned in the command output:

Authentication Failures
  Specifies the number of times a requesting principal failed authentication procedures.

Cache Bypasses
  Specifies the number of requests to read attributes for which the CDS clerk was specifically directed by the requesting application to bypass its own cache. Instead, a CDS server is contacted to get the requested information. This attribute does not account for requests that the CDS clerk is unable to satisfy from the cache or for requests to look up names or enumerate the contents of directories.

Cache Hits
  Specifies the total number of read requests directed to the CDS clerk that were satisfied entirely by the information contained in its own cache. This attribute accounts only for requests to read attribute values and does not include requests to look up names or enumerate the contents of directories.

Creation Time
  Specifies the time when this entity was created.

Miscellaneous Operations
  Specifies the number of operations other than read and write, such as skulks and enumerating contents of directories, performed by the CDS clerk.

Read Operations
  Specifies the number of lookup operations performed by the CDS clerk. This attribute accounts only for requests to read attributes and does not include requests to look up names or enumerate the contents of directories.

Write Operations
  Specifies how many requests to modify data were processed by the CDS clerk.

Privilege Required: You must have r (read) permission to the clerk.

Examples

The following command displays the attributes of the CDS clerk on the local system:
show clerk

cdscp> show clerk

SHOW
CLERK
Authentication failures = 0
Read Operations = 1068
Cache Hits = 137
Cache Bypasses = 433
Write Operations = 1250
Miscellaneous Operations = 590

Related Information

Commands:

disable clerk

Books:

- z/OS DCE Administration Guide
show directory

Displays attribute information about the specified directory.

Format


Parameters

directory-name The full name of a specific directory. The last simple name can contain wildcard characters.

attribute-name The name of a particular attribute; see the Attributes section below for valid attribute names. An application-defined attribute is also valid here. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes that your application uses.

attribute-value The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in “Attributes” on page 525.

Usage

The show directory command displays the names and values of the attributes specified in attribute-name. You can use a combination of attributes in any sequence in a single command. Use a space to separate multiple attributes. You can use a with attribute-name [=] attribute-value clause to limit output only to directories whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=). If you do not supply any attributes, the command displays all attributes and their values. In addition to the directory attributes listed below, application-specific attributes can exist for a directory.

Attributes

CDS_AllUpTo Indicates the date and time of the last successful skulk on the directory. All replicas of the directory are guaranteed to receive all updates whose timestamps are less than the value of this attribute. Data_type = Timestamp.

CDS_Convergence Specifies the degree of consistency among replicas. The value of this attribute is defined as one of the following:

- **low** CDS does not immediately propagate an update. The next skulk distributes all updates that occurred since the previous skulk. Skulks occur at least once every 24 hours.

- **medium** CDS attempts to immediately propagate an update to all replicas. If the attempt fails, the next scheduled skulk makes the replicas consistent. Skulks occur at least once every 12 hours.

- **high** CDS attempts to immediately propagate an update to all replicas. If the attempt fails (for example, if one of the replicas is unavailable), a skulk is scheduled for within 1 hour. Skulks usually occur at least once every 12 hours. Use this setting temporarily and briefly, because it uses extensive system resources.
show directory

By default, every directory inherits the convergence setting of its parent at creation time. The default setting on the root directory is medium. Data_type = Character.

**CDS_CTS**
Specifies the creation timestamp (CTS) of the CDS directory. Data_type = Timestamp.

**CDS_DirectoryVersion**
Specifies the current version of the directory (derived from the CDS_DirectoryVersion attribute of the clearinghouse in which the directory was created). Multiple directory versions are supported in a cell. Data_type = Version.

**CDS_Epoch**
Specifies a UUID that identifies a particular incarnation of the directory. Data_type = UUID.

**CDS_InCHName**
Indicates whether a directory or any of its descendants can store clearinghouse names.

If this value is true
The directory can store clearinghouse names

If this value is false
The directory cannot store clearinghouse names

Data_type = Character.

**CDS_LastSkulk**
Records the timestamp of the last skulk performed on this directory. Data_type = Timestamp.

**CDS_LastUpdate**
Records the timestamp of the most recent change to any attribute of a directory replica, or any change to an entry in the replica. Data_type = Timestamp.

**CDS_ObjectUUID**
Specifies the unique identifier of the directory. Data_type = UUID.

**CDS_ParentPointer**
Contains a pointer to this directory’s parent in the name space. Data_type = ParentPointer

**CDS_Replicas**
Specifies the address, UUID, and name of every clearinghouse where a copy of this directory is located. This attribute also specifies whether the replica in a particular clearinghouse is a master or read-only replica. CDS_Replicas is not a data type; it is a set of data types like a structure.

**CDS_ReplicaState**
Specifies whether a directory replica can be accessed. Data_type = Small

**CDS_ReplicaType**
Indicates whether a directory replica is a master or read-only replica. Data_type = Small

**CDS_ReplicaVersion**
Specifies the version of a replica of the directory. Data_type = Version.

**CDS_UTS**
Specifies the timestamp of the most recent update to an attribute of the directory. Data_type = Timestamp.

**RPC_ClassVersion**
Specifies the RPC runtime software version that can be used to import on the directory. Data_type = Byte.
Privilege Required: You must have r (read) permission to the directory. If you specify a wildcard directory name, you also must have r (read) permission to the parent directory of the directory.

Note: The show directory command is supported in z/OS DCE by the dcecp directory show command.

Examples

The following command displays the current values of all the attributes associated with the ./admin directory:

cdscp> show directory ./admin

SHOW
DIRECTORY /.../abc.com/admin
RPC_ClassVersion = 0100
  CDS_STS = 1996-10-15-13:09:47.000000003/08-00-2b-1c-8f-1f
  CDS_UTS = 1996-10-17-08:59:50.000000006/08-00-2b-1c-8f-1f
  CDS_ObjectUUID = ba700c98-8b1a-11ca-8981-08002b0f79aa
  CDS_Replicas = :
  Clearinghouse_UUID = 2ab024a8-8b1a-11ca-8981-08002b0f79aa
    Tower = ncadg_ip_udp:16.20.16.32
    Tower = ncadg_ip_tcp:16.20.16.32
    Replica type = master
    Clearinghouse_Name = /.../abc.com/Paris_CH
    CDS_AllUpTo = 1996-10-17-08:51:18.000000032/08-00-2b-1c-8f-1f
    CDS_Convergence = medium
    CDS_ParentPointer = :
      Parent's_UUID = b773525c-8b1a-11ca-8981-08002b0f79aa
      Timeout = :
      Extension = t1-00:00:00.000
      MyName = /.../abc.com/admin
    CDS_DirectoryVersion = 3.0
    CDS_ReplicaState = on
    CDS_ReplicaType = master
    CDS_LastSkulk = 1996-10-17-08:51:18.000000032/08-00-2b-1c-8f-1f
    CDS_LastUpdate = 1996-10-21-13:04:02.000000044/08-00-2b-1c-8f-1f
    CDS_Epoch = bd8b2c50-8b1a-11ca-8981-08002b0f79aa
    CDS_ReplicaVersion = 3.0

Related Information

Commands:

add directory delete directory remove directory
create directory list directory set directory

Books:

• z/OS DCE Administration Guide
show link

Displays attribute information about the specified soft link.

Format


Parameters

link-name        The full name of a specific soft link. The last simple name can contain wildcard characters.
attribute-name   The name of a particular attribute; see the Attributes section below for valid attribute names.
attribute-value  The value of a particular attribute. Information about the formats required for entering values of various data types is found in "Attributes" on page 525.

Usage

The show link command displays the names and values of the attributes specified in attribute-name. You can use a combination of attributes in any sequence in a single command. Use a space to separate multiple attributes. You can use a with attribute-name [=] attribute-value clause to limit output only to soft links whose attributes have values equal to the specified values. A space must precede and follow the equal sign (=). If you do not supply any attributes, the command displays all attributes and their values.

Attributes

CDS_CTS          Specifies the creation timestamp (CTS) of this soft link. Data_type = Timestamp.
CDS_LinkTarget   Specifies the full name of the directory, object entry, or other soft link to which the soft link points. Data_type = Character.
CDS_LinkTimeout  Specifies a timeout value after which the soft link is either checked or deleted. Data_type = Timeout.
CDS_UTS          Specifies the timestamp of the most recent update to an attribute of the soft link. Data_type = Timestamp.

Privilege Required:  You must have r (read) permission to the soft link. If you specify a wildcard soft link name, you also must have r (read) permission to the directory that stores the soft link.

Note:  The show link command is supported in z/OS DCE by the dcecp link show command.

Examples

The following command displays the current values of all the attributes associated with the soft link /./sales/region1:
cdscp> show link /.:sales/region1

SHOW
SOFTLINK  /.../abc.com/sales/region1
CDS_CTS = 1996-10-15-19:54:36.00000077/08-00-2b-1c-8f-1f
CDS_UTS = 1996-10-15-19:54:36.00000009/08-00-2b-1c-8f-1f
CDS_LinkTarget = /.../abc.com/sales/software
CDS_LinkTimeout = :
Expiration = 1996-10-15-00:00:00.0
Extension = +1-00:00:00.000

Related Information

Commands:
create link    list link    set link
delete link    remove link

Books:
• [z/OS DCE Administration Guide]
show object

Displays attribute information about the specified object entry.

Format


Parameters

object-name The full name of a specific object entry. The last simple name can contain wildcard characters.

attribute-name The name of a particular attribute; see the Attributes section below for valid attribute names. An application-defined attribute is also valid here. See the cds_attributes file (/opt/dcelocal/etc/cds_attributes) for the list of attributes that your application uses.

attribute-value The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute. See the cds_attributes file for the list of attributes and corresponding data types that your application uses. Information about the formats required for entering values of various data types is found in "Attributes" on page 525.

Usage

The show object command displays the names and values of the attributes specified in attribute-name. You can use a combination of attributes in a single command. A space should separate multiple attributes as well as precede and follow the equal sign (=). You can use a with attribute-name [=] attribute-value clause to limit output to object entries whose attributes have values equal to the specified values. If you do not supply any attributes, the command displays all attributes and their values. In addition to the attributes listed below, any application-defined attributes that might exist will be included in the output of this command.

Attributes

CDS_Class Specifies the class to which an object belongs. Data_type = Character.

CDS_CLASSVersion Contains the version number of the object’s class. This allows applications to build in compatibility with entries created by earlier versions. Data_type = Version.

CDS_CTS Specifies the creation timestamp (CTS) of this object entry. Data_type = Timestamp.

CDS_ObjectUUID Specifies a unique identifier for the object being referenced. Data_type = UUID.

CDS_UTS Specifies the timestamp of the most recent update to an attribute of the object entry. Data_type = Timestamp.

Privilege Required: You must have r (read) permission to the object entry. If you specify a wildcard object entry name, you also must have r (read) permission to the directory that stores the object entry.

Note: The show object command is supported in z/OS DCE by the dcecp object show command.
Examples

The following command lists all the attributes and their values of the object entry /./sales/east/floor1cp:

cdscp> show object /./sales/east/floor1cp

SHOW
OBJECT      /.../abc.com/sales/east/floor1cp
CDS_CTS      = 1996-10-15-19:53:03.00000003/08-00-2b-1c-8f-1f
CDS_UTS      = 1996-10-15-19:53:03.00000006/08-00-2b-1c-8f-1f
CDS_ObjectUUID = 1996-10-15-19:53:03.00000006/08-00-2b-1c-8f-1f

Related Information

Commands:

- add object
- create object
- delete object
- list object
- remove object
- set object

Books:

- [z/OS DCE Administration Guide](#)
show replica

Displays attribute information about the specified replica.

Format

cdscp sh[ow] r[eplica] directory-name cl[earinghouse] clearinghouse-name [attribute-name]

Parameters

directory-name   The full name of the directory.
clearinghouse-name The full name of the clearinghouse. The clearinghouse name must be in the cell.
attribute-name    The name of a particular attribute; see the Attributes section below for valid attribute names.

Usage

The **show replica** command displays the directory-specific attributes as well as the per-replica attributes of the specified directory. You can use a combination of attributes in any sequence in a single command. Use a space to separate multiple attributes. If you do not supply any attributes, the command displays all attributes and their values. Existing application-defined attributes are included in the output of this command.

Attributes

**CDS_AllUpTo** Indicates the date and time of the last successful skulk on the directory. All replicas of the directory are guaranteed to have received all updates whose timestamps are less than the value of this attribute. Data_type = Timestamp.

**CDS_Convergence** Specifies the degree of consistency among replicas. The value of this attribute is defined as one of the following: Data_type = Character.

- **low** CDS does not immediately propagate an update. The next skulk distributes all updates that occurred since the previous skulk. Skulks occur at least once every 24 hours.
- **medium** CDS attempts to immediately propagate an update to all replicas. If the attempt fails, the next scheduled skulk makes the replicas consistent. Skulks occur at least once every 12 hours.
- **high** CDS attempts to immediately propagate an update to all replicas. If the attempt fails (for example, if one of the replicas is unavailable), a skulk is scheduled for within 1 hour. Skulks usually occur at least once every 12 hours. Use this setting temporarily and briefly, because it uses extensive system resources.

By default, every directory inherits the convergence setting of its parent at creation time. The default setting on the root directory is **medium**.

**CDS_CTS** Specifies the creation timestamp (CTS) of the directory of which this replica is a copy. Data_type = Timestamp.
show replica

**CDS_DirectoryVersion**
Specifies the current version of the directory (derived from the CDS_DirectoryVersion attribute of the clearinghouse in which the directory was created.) Data_type = Version.

**CDS_Epoch**
Specifies a UUID that identifies a particular instance of the directory. Data_type = UUID.

**CDS_InCHName**
Indicates whether a directory or any of its descendants can store clearinghouse names.
If this value is true
The directory can store clearinghouse names
If this value is false
The directory cannot store clearinghouse names
Data_type = Character.

**CDS_LastSkulk**
Records the timestamp of the last skulk performed on this particular replica of a directory. Data_type = Timestamp.

**CDS_LastUpdate**
Records the timestamp of the last update to any attribute of the replica, or any change to the contents of the replica, including object entries, child pointers, and soft links. Data_type = Timestamp.

**CDS_ObjectUUID**
Specifies the unique identifier of the directory of which this replica is a copy. Data_type = UUID.

**CDS_ParentPointer**
Contains a pointer to this directory’s parent in the namespace. Data_type = ParentPointer.

**CDS_Replicas**
Specifies the address, UUID, and name of every clearinghouse where a replica of this directory is located. This attribute also specifies whether the replica in a particular clearinghouse is a master or read-only replica. CDS_Replicas is not a data type; it is a set of data types like a structure.

**CDS_ReplicaState**
Specifies the internal state of a replica. When you create or delete a replica, it goes through various states. Data_type = Small. Possible values are:
1. new directory
2. new replica
3. on
4. dying replica
5. dying directory
6. dead

**CDS_ReplicaType**
Specifies the replica type of a directory. This could be a master or read-only replica. Data_type = Character.

**CDS_ReplicaVersion**
Specifies the replica version of a directory. Data_type = Version.

**CDS_UTS**
Specifies the timestamp of the most recent update to an attribute of the directory. Data_type = Timestamp.

**RPC_ClassVersion**
Specifies the RPC runtime software version that can be used to import on the directory. Data_type = Byte.
show replica

Privilege Required: You must have r (read) permission to the directory from which the replica is created.

Note: The show replica command is supported in z/OS DCE by the dcecp directory show command.

Examples

The following command displays the current values of all the attributes of the replica of the /.:/eng directory in the /.:/Chicago2_CH clearinghouse:

cdscp> show replica /.:/eng clearinghouse /.:/Chicago2_CH

SHOW REPLICA /.../abc.com/eng
RPC_ClassVersion = 0100
    CDS_CTS = 1996-10-15-12:09:47.000000003/08-00-2b-1c-8f-1f
    CDS_UTS = 1996-10-17-07:59:50.000000006/08-00-2b-1c-8f-1f
    CDS_ObjectUUID = 5816da70-8b1c-11ca-8981-08002b0f79aa
    CDS_Replacics = :
    Clearinghouse UUID = 2ab024a8-8b1a-11ca-8981-08002b0f79aa
        Tower = ncadg_ip_udp:16.20.16.32
        Tower = ncadg_ip_tcp:16.20.16.32
    Replica type = master
    Clearinghouse Name = /.../abc.com/Chicago1_CH
    CDS_Replacics = :
    Clearinghouse UUID = 49757f28-8b1a-11ca-8981-08002b0f79aa
        Tower = ncadg_ip_udp:16.20.16.32
        Tower = ncadg_ip_tcp:16.20.16.32
    Replica type = readonly
    Clearinghouse Name = /.../abc.com/Chicago2_CH
    CDS_AllUpTo = 1996-10-17-07:51:18.000000032/08-00-2b-1c-8f-1f
    CDS_Convergence = medium
    CDS_ParentPointer = :
        Parent's UUID = 560f1ad0-8b1c-11ca-8981-08002b0f79aa
        Timeout = :
        Extension = +1-00:00:00.000
    CDS_DirectoryVersion = 3.0
    CDS_ReplacicaState = on
    CDS_ReplacicaType = readonly
    CDS_LastSkulk = 1996-10-17-07:51:18.000000032/08-00-2b-1c-8f-1f
    CDS_LastUpdate = 1996-10-21-12:04:02.000000044/08-00-2b-1c-8f-1f
    CDS_Epoch = 58472144-8b1c-11ca-8981-08002b0f79aa
    CDS_ReplacicaVersion = 3.0

Related Information

Commands:
create replica  set directory to new epoch  delete replica

Books:
• z/OS DCE Administration Guide
show server

Displays attribute information about the server running on the local system.

Format

cdscp sh[ow] se[rv]er

Usage

The `show server` command displays all the names and values from the attributes named in this entity. The server must be enabled when you use this command.

Attributes

Child Update Failures
Specifies the number of times the server was unable to contact all the clearinghouses that store a replica of a particular child directory’s parent directory and apply the child updates that occurred since the last skulk. This counter is incremented by the Cannot Update Child Pointer event.

Creation Time
Specifies the time when the cdscp process was started.

Crucial Replicas
Specifies the number of times a user attempted (from this server) to remove a replica that is crucial to the connectivity of a directory hierarchy. The server background process prevents users from accidentally disconnecting lower-level directories from higher-level directories. When it detects an attempt to remove a crucial replica, it does not process the command to do so. This counter is incremented by the Crucial Replica event.

Future Skew Time
Specifies the maximum amount of time that a timestamp on a new or modified entry can vary from local system time at the server system.

Known Clearinghouses
Specifies the clearinghouse or clearinghouses known to the server.

Read Operations
Specifies the number of read operations directed to this Cell Directory Service (CDS) server.

Security Failures
Specifies the number of times a server principal for this server had inadequate permission to perform a requested operation.

Skulks Completed
Specifies the number of skulks successfully completed by this CDS server.

Skulks Initiated
Specifies the number of skulks started by this CDS server.

Times Lookup Paths Broken
Specifies the number of broken connections between clearinghouses on this server and clearinghouses closer to the root. Incoming requests to this server that require a downward lookup in the directory hierarchy still succeed, but requests that require a lookup in directories closer to the root will fail. This counter is incremented by the Broken Lookup Paths event.
show server

Write Operations
Specifies the number of write operations to this CDS server.

Privilege Required: You must have r (read) permission to the server.

Examples
The following command displays the current values of all the attributes associated with the server running on the local system:

cdscp> show server
SHOW SERVER AT
= 757 Write Operations = 542 Skulks Initiated = 219 Skulks Completed =
219 Times Lookup Paths Broken = 1 Crucial Replicas = 0 Child Update
Failures = 1 Security Failures = 0 Known Clearinghouses =
/.../abc.com/Boston_CH = /.../abc.com/NY_CH

Related Information
Commands:
disable server
Chapter 5. Distributed Time Service

This chapter introduces you to the DCE Distributed Time Service (DTS), and describes the DTS control program (DTSCP) administrative facilities and their associated commands. The DTS daemon (dtsd) and the DTS null time provider program, dtstp, are started by the modify system operator command. For further information about the modify command, see Chapter 1, “DCE and z/OS Commands” on page 1.

The DTS control program is started by the dtscp command. It sends administration commands to the DTS daemon, which synchronizes, adjusts, and maintains the DCE software clocks in a distributed network. These software clocks are visible only through the DTS entity. dtscp management commands enable the administrator to communicate with the DTS daemon.

You can configure the DTS entity as either a server or as a clerk.

The following DTSCP command is not supported in z/OS DCE:

- dtsdate

Related Information

Commands (PROCs or CLIST):

dtscp
advertise

advertise

Configures the system as a global server by adding the server’s entry to the cell profile.

Format

dtscp adv[ertise]

Usage

The advertise command causes DTS to forward the name and attributes of the server to CDS by binding the server’s protocol tower to the CDS object and adding an entry for the server in the cell profile. When the server’s entry is in the cell profile, it is configured as a global server, and servers outside of the local set can access it.

Privilege Required: You must have write permission on the ACL associated with the DTS entity to process the command.

Note: The advertise command is supported in z/OS DCE by the dcecp dts configure command.

Examples

dtscp> advertise

Related Information

Commands:

unadvertise
change

Alters the epoch number and time on the local node.

Format

dtscp change [epoch] integer [time] absolute-time

Parameters

epoch integer Specifies the new epoch number (0 - 255). This argument is required.
time absolute-time Specifies a clock setting for the new epoch. If you do not supply this argument and a value, the server uses the current clock time with an unspecified inaccuracy and initiates a synchronization. This argument is optional.

Usage

The change command sets the time and changes the epoch of the DTS server on which it is entered. Use this command to isolate a server from the rest of the servers in the network before changing the time.

Privilege Required: You must have write permission on the ACL associated with the DTS entity to process this command.

Notes:

1. This command is valid only for servers. The new epoch number you specify must be different from the current epoch number.
2. The change command is supported in z/OS DCE by the dcecp dts modify command.

Examples

1. This example shows how to change the epoch number:
   dtscp> change epoch 1
2. This example shows how to change the epoch number and time:
   dtscp> change epoch 1 time 1996-11-30-10:58:00.000-05:0010.000

Related Information

Commands:

update
create

Creates the DCE DTS entity on the specified node.

Format

dtscp create type

Parameters

type  Specifies the type of DCE DTS entity to be created on the specified node. Specify one of the following for type. If no argument is specified, the system defaults to clerk.
clerk  The DCE DTS entity is created as a clerk.
server  The DCE DTS entity is created as a server.

Usage

The create command creates a DTS server or DTS clerk entity on the system where the command is entered.

After the DTS entity is created, it is still in a non-functioning state. Use the dcecp dts activate or dtscp enable command to move this entity into a functioning-state, and cause immediate synchronization. See "enable" on page 619 for more information.

Privilege Required:  You must have write permission on the ACL associated with the DTS entity to process this command.

Examples

dtscp> create type server
delete

Deletes the DCE DTS entity.

Format

dtscp del[ete]

Usage

The delete command deletes the DCE DTS entity from the system where the command is entered. When delete is processed, the DTS daemon completes processing. To restart the DTS daemon, use the z/OS modify command.

Privilege Required: You must have write permission on the ACL associated with the DTS entity to process the command.

Notes:

1. The DCE DTS entity cannot be deleted until you enter the disable command, which causes the status attribute state to be set from on to off.
2. The delete command is supported in z/OS DCE by the dcecp dts stop command.

Examples

dtscp> delete

Related Information

Commands:

disable
disable

Stops the DCE DTS entity on the local node.

Format

dtscp dis[able]

Usage

The disable command turns off the DCE DTS entity on the system where the command is entered. When the command is processed, the status attribute state is set from on to off. The disable command stops the DTS entity from synchronizing, adjusting, and maintaining the system clock.

Privilege Required: You must have write permission on the ACL associated with the DTS entity to process this command.

Notes:

1. The DCE DTS entity cannot be disabled until it is enabled with the dcecp dts activate or dtscp enable command. You must enter the dcecp dts deactivate or dtscp disable command before you can delete the entity.

2. The disable command is supported in z/OS DCE by the dcecp dts deactivate command.

Examples

dtscp> disable

Related Information

Commands:

enable delete create
DTS Control Program

The DTS control program is a command-line interface. It sends administration commands to the DTS
daemon, which synchronizes, adjusts, and maintains the DCE software clocks in a distributed network.
These software clocks are visible only through the DTS entity. The DTS daemon does not update the
hardware system clocks. For a detailed explanation of system clock synchronization and management,
see the z/OS DCE Administration Guide.

You can use control program commands from within the control program or from the TSO system prompt.
To enter DTS commands from within the control program, first start the control program by entering the
dtscp command as follows:

READY

dtscp

dtscp>

At this prompt, you can enter any control program command, such as:

dtscp> show current time

To leave the control program and return to the system prompt, enter the exit or quit command.

To enter DTS commands from the system prompt (interactively or in a command procedure), enter the
dtscp command with a command of the control program as the first argument. The control program
processes the command without displaying the control program prompt. For example, you can enter the
synchronize command as follows:

dtscp synchronize

Enter the set server entry [name], set server group [name], and set server principal [name] dtscp
commands from native TSO/E or the shell only. If these commands and their arguments are entered from
the ISPF command line, they are converted to lowercase characters. This is a problem for arguments that
have mixed or all uppercase characters.

Some dtscp commands have optional arguments or attributes; there may also be optional variables for
the arguments and attributes.

If you run dtscp in batch mode, the input to the control programs and the output for each command are
written to the job execution log. You can then determine which input commands were successful and
which were not.
dtscp

The Distributed Time Service (DTS) control program.

Format

dtscp

Note: With the exception of the following subcommands, this command is supported in z/OS DCE by the dcecp command.

- create
- exit
- help
- quit

Usage

This section describes the commands for the DCE DTS control program (dtscp). The DTS control program is a command-line interface that enables you to synchronize, adjust, and maintain the system clocks in a distributed network. For a detailed explanation of system clock synchronization and management, see the z/OS DCE Administration Guide.

DTS Control Program Commands

The DTS control program commands are:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>advertise</td>
<td>Configures the DTS server as a global server</td>
</tr>
<tr>
<td>change</td>
<td>Modifies the epoch and sets the local time to a new time</td>
</tr>
<tr>
<td>create</td>
<td>Establishes a DTS entity (a clerk or server)</td>
</tr>
<tr>
<td>delete</td>
<td>Causes DTS to exit on the local node</td>
</tr>
<tr>
<td>disable</td>
<td>Suspends a DTS entity</td>
</tr>
<tr>
<td>enable</td>
<td>Activates a DTS entity</td>
</tr>
<tr>
<td>exit</td>
<td>Ends the dtscp management session and returns you to the system prompt</td>
</tr>
<tr>
<td>help</td>
<td>Starts the dtscp help service</td>
</tr>
<tr>
<td>quit</td>
<td>Ends the dtscp management session and returns you to the system prompt</td>
</tr>
<tr>
<td>set</td>
<td>Modifies characteristics of a DTS entity</td>
</tr>
<tr>
<td>show</td>
<td>Displays characteristics and status of a DTS entity</td>
</tr>
<tr>
<td>synchronize</td>
<td>Synchronizes the system clock with the time obtained from DTS servers in the network</td>
</tr>
<tr>
<td>unadvertise</td>
<td>Removes the global server entry</td>
</tr>
<tr>
<td>update</td>
<td>Gradually adjusts the system clock to a new time</td>
</tr>
</tbody>
</table>

For more information on any of the above dtscp commands, see the appropriate page in this chapter.

The following table shows the dcecp commands that replace the dtscp commands:
Table 15. DCECP Equivalent Commands for DTSCP Commands

<table>
<thead>
<tr>
<th>dtscp Command</th>
<th>dcecp Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>dtscp</td>
<td>dcecp</td>
</tr>
<tr>
<td>advertise</td>
<td>dts configure</td>
</tr>
<tr>
<td>change</td>
<td>dts modify</td>
</tr>
<tr>
<td>create</td>
<td>None</td>
</tr>
<tr>
<td>delete</td>
<td>dts stop</td>
</tr>
<tr>
<td>disable</td>
<td>dts deactivate</td>
</tr>
<tr>
<td>enable</td>
<td>dts activate</td>
</tr>
<tr>
<td>exit</td>
<td>exit</td>
</tr>
<tr>
<td>help</td>
<td>dts help</td>
</tr>
<tr>
<td>quit</td>
<td>quit</td>
</tr>
<tr>
<td>set</td>
<td>dts modify</td>
</tr>
<tr>
<td>show</td>
<td>dts show</td>
</tr>
<tr>
<td>synchronize</td>
<td>dts synchronize</td>
</tr>
<tr>
<td>unadvertise</td>
<td>dts configure</td>
</tr>
<tr>
<td>update</td>
<td>clock set</td>
</tr>
</tbody>
</table>

You can use control program commands from within the control program or from the system prompt. To enter DTS commands from within the control program, first start the control program by entering the `dtscp` command. For example:

```
$ dtscp
dtscp>
```

At this prompt you can enter any control program command. For example:

```
dtscp> show current time
```

To leave the control program and return to the system prompt, enter the `exit` command.

To enter DTS commands from the system prompt (interactively or in a command procedure), enter the `dtscp` command with an internal command of the control program as the first argument. The control program processes the command without displaying the control program prompt. For example, you can enter the `synchronize` command as follows:

```
$ dtscp synchronize
```

Some `dtscp` commands have optional arguments or attributes; there may be optional variables for the arguments and attributes.

See the following sample command:

```
dtscp> update time 1990-08-03 05:45:28.000+01:00 I00.500
```

The `Format` section of each command shows exactly how that command can be abbreviated. The characters that need not be entered are enclosed in square brackets.
**Attribute Groups:** The `dtscp show` command lets you specify attributes—pieces or sets of data. These attributes are described more fully in the sections which deal with the `show` command. The `show` command also lets you specify attribute groups, which are groups of individual attributes. The valid attribute groups are described in the following list:

**Characteristics** Set or show the entity's operation.

**Counters** Show the number of occurrences of an event since the entity was enabled.

**Status** Show the current state of the entity. (The DTS entity has four status attributes.)

**Global Servers** Show the global servers known by this DTS entity.

**Local Servers** Show the local servers known by this DTS entity.

Individual attributes within each of the previously listed groups are described in the pages for the `set` and `show` commands. The `show` command also lets you specify attribute groups.

**Time Stamps:** All responses to commands contain a timestamp. The following example shows a typical DTS time display:

```
1996-03-16-14:29:47.52000-05:001000.003
```

The timestamp uses the DTS format that is explained in the chapter on DTS concepts in the **Z/OS DCE Administration Guide**. In this example, the year is 1996, the day is March 16, and the time is 14 hours, 29 minutes, and 47.52 seconds. A negative Time Differential Factor (TDF) of 5 hours and an inaccuracy of 3 milliseconds are included in the timestamp.

**Note:** An inaccuracy value of `I-----` indicates an infinite inaccuracy. This value appears in time displays before a node's initial synchronization or after you enter the `change` command without specifying an inaccuracy value.

**Related Information**

Commands:

- `advertise`
- `change`
- `create`
- `disable`
- `enable`
- `exit`
- `help`
- `set`
- `show`
- `synchronize`
- `quit`
- `unadvertise`
- `update`
enable

Starts the DCE DTS entity on the local node.

Format

dtscp enable [set clo[ck] boolean]

Parameters

[set clock boolean]

Specifies whether the clock is abruptly set or gradually adjusted to the computed time. This argument is optional. Valid values for boolean are:

- false: The clock is gradually adjusted. This is the default condition.
- true: The clock is abruptly set.

Usage

The enable command starts the DCE DTS entity on the system where the command is entered. To put the entity into operation you must use the enable command, which causes an immediate synchronization to take place. When the command is run, the status attribute state is set from off to on.

Use the enable command to activate a DTS entity that was previously deactivated with the dcecp dts deactivate or dtscp disable command. See "disable" on page 614 for more information.

Privilege Required: You must have write permission on the ACL associated with the DTS entity to run the command.

Notes:

1. The DCE DTS entity cannot be enabled until it is created with the create command; the DTS entity must be in the off state to be enabled.
2. The enable command is supported in z/OS DCE by the dcecp dts activate command.

Examples

1. This example shows how to enable the entity and adjust the clock gradually to the computed time following the first synchronization:

   dtscp> enable

2. This example shows how to enable the entity and abruptly set the clock to the computed time following the first synchronization:

   dtscp> enable set clock true

Related Information

Commands:

create disable
exit

Causes the DCE DTS control program (DTSCP) to complete execution.

**Format**

```
dtscp exit[t]
```

**Usage**

The `exit` command causes the DTSCP to complete processing and returns control to the parent process.

**Privilege Required:** None required.

**Examples**

The following command shows how to leave the DTSCP and return to the parent process.

```
dtscp> exit
```

**Related Information**

Commands:

- `quit`
help

Displays help information about DCE dtscp commands.

Format

```
dtscp help [help-topic]
```

Parameters

```
help-topic
```

Specifies the help topic for which help information is to be displayed. The following are valid help topics:

- adv[ertise]
- dis[able]
- syn[chronize]
- cha[nge]
- ena[ble]
- una[dvertise]
- cre[ate]
- set
- upd[ate]
- del[ete]
- sho[w]

Usage

The help command displays information about DTSCP commands.

Privilege Required: None required.

Note: The help command is supported in z/OS DCE by the dcecp dts help command.

Examples

The following command shows how to get help about the help subtopic unadvertise:

```
dtscp> help unadvertise
```
**quit**

Causes the DCE `dtscp` control program to complete processing.

**Format**

`dtscp qui[t]`

**Usage**

The `quit` command causes the `dtscp` to complete processing and returns control to the parent process.

**Privilege Required:** None required.

**Examples**

The following command shows how to leave the DTSCP and return to the parent process.

```
dtscp> quit
```

**Related Information**

Commands:

`exit`
set

Modifies characteristics for the DCE DTS entity.

Format

dtscp set characteristic

Parameters

characteristic The name and value of one or more characteristics to be modified. Valid values for characteristic are described in the following list. These values are described in more detail in the Usage section.

- check [interval] [relative-time]
- courier [role] [role]
- error [tolerance] [relative-time]
- global set [timeout] [relative-time]
- local set [timeout] [relative-time]
- maximum [inaccuracy] [relative-time]
- query [attempts] [integer]
- server entry [name] [name]
- server group [name] [name]
- server principal [name] [name]
- servers required [integer]
- synchronization [hold [down]] [relative-time]

Usage

The set command modifies the characteristics specified for the DTS entity. The variable of the characteristic argument has an assigned default and value range and can take up to 80 characters.

The characteristic argument variable relative time is particular in that it will be recalculated to the correct date format. For example:

Input Value => 0-0025:10:99.99999999
Result => 1-01:11:39.990

The assigned default and value range numbers for this variable will be read as follows:

    X-XX:XX:XX.XXX
\   \   \   \   \   \   
    days \ min \ seconds (with 3 decimal places) \ hours

Note: The rule for a three-character abbreviation (as discussed in the description for "DTS Control Program" on page 615) holds true for all of these commands.
The modifiable characteristics and their values are described in the following list.

**check [interval]**  Specifies the amount of time between checks for faulty servers. Applicable only for servers that have external time providers.

Default: 0-01:30:00.000

Value: 0-00:00:30.000 - 10675199-02:48:05.000

The value indicates a range from 30 seconds to infinity.

**courier [role]**  Specifies a server’s interaction with the set of global servers.

Default: Backup courier

Value:

- **backup courier**  The local server becomes a courier if none is available in the local set.
- **courier**  The local server synchronizes with the global set of servers.
- **noncourier**  The local server does not synchronize with the global set of servers.

**error [tolerance]**  Specifies the maximum separation allowed between the local clock and the computed time before synchronizations become abrupt rather than gradual (monotonic).

Default: 0-00:10:00.000

Value: 0-00:00:00.500 - 10675199-02:48:05.000

**global set [timeout]**  Specifies the amount of time the node waits for a response to a global synchronization request before sending another request or declaring a global server unavailable. The node sends out a number of query-attempt requests before declaring a server unavailable that is not responding. The number of query attempts is controlled by the query-attempts characteristic. The default number of query attempts is three.

Default: 0-00:01:30.000

Value: 0-00:00:00.000 - 0-00:10:00.000

**local set [timeout]**  Specifies the amount of time the node waits for a response to a local synchronization request before sending another request or declaring a server unavailable. The node sends out a number of query-attempt requests before declaring a server unavailable that is not responding. The number of query attempts is controlled by the query-attempts characteristic. The default number of query attempts is three. The **local set timeout** value controls only the initial contact with a time provider. During this initial contact, the time provider determines the timeout value for reporting back time, allowing a time provider attached to a slow source, like a modem, to request that **dtsd** wait for a longer interval.

Default: 0-00:01:00.000

Value: 0-00:00:00.000 - 0-00:02:00.000

**maximum [inaccuracy]**  Specifies the inaccuracy limit for the node. When the node exceeds the maximum inaccuracy setting, it attempts to synchronize. The maximum inaccuracy value must be less than 24 hours.

Default: 0-00:00:00.100

Value: 0-00:00:00.0 - 24:00:00.0
query [attempts]
Specifies the number of attempts that a node makes to contact a server before the node considers the server unavailable.
Default: 3
Value: 1 - 10

server entry [name]
Specifies a server’s CDS entry name; <hostname> represents the name of the system or node that is the server’s client. The default setting is the recommended value.
Default: hosts/<hostname>/dts-entity
Value: The default setting.

server group [name]
Specifies the security group name that DTS uses for authentication checks. DTS clerks and servers do not accept time values from DTS servers that are not included in this group. The default setting is the recommended value.
Default: subsys/dce/dts-servers
Value: The default setting.

server principal [name]
Specifies a server’s principal name for authentication purposes; <hostname> represents the name of the system or node that is the server’s client. The default setting is the recommended value.
Default: hosts/<hostname>/self
Value: The default setting.

servers required
Specifies the minimum number of servers required for a synchronization. Settings of 1 or 2 may cause unreliable computed times.
Default: 1 (clerks) 3 (servers)
Value: 1-32

synchronization [hold [down]]
Specifies the interval a node must wait to synchronize.
Clerks:
Default: 0-00:10:00.000
Value: 0-00:00:30.000 - 01-00:00:00.000
Servers:
Default: 0-00:02:00.000
Value: 0-00:00:30.000 - 01-00:00:00.000

Privilege Required: You must have write permission on the ACL associated with the DTS entity to run the command.

Note: The set command is supported in z/OS DCE by the dcecp dts modify command.

Examples
1. The following example command sets the check interval characteristic to 30 seconds:
set

dtscp> set check interval 00:00:00:30.000

2. This example shows how to set the number of servers required before the entity can synchronize:
dtscp> set servers required 4

3. This example shows how to set the courier role for a server:
dtscp> set courier role backup courier

4. This example command sets the error tolerance characteristic to 7 minutes:
dtscp> set error tolerance
00:00:07:00.000

5. This example command sets the global set timeout characteristic to 45 seconds:
dtscp> set global set timeout
00:00:45:000

6. This example command sets the local set timeout characteristic to 5 seconds:
dtscp> set local set timeout
00:00:05:000

7. This example command sets the maximum inaccuracy characteristic to 3 milliseconds:
dtscp> set maximum inaccuracy
00:00:00:300

8. This example command sets the server entry name characteristic to /:/hosts/orion/dts-entity:
dtscp> set server entry name
hosts/orion/dts-entity

9. This example command sets the server principal name characteristic to /:/hosts/vega/dts-entity:
dtscp> set server principal name
hosts/vega/dts-entity

   Note: In z/OS, the principal name is a read-only attribute and cannot be modified.

10. This example command sets the synchronization hold-down characteristic to 15 minutes:
dtscp> set synchronization hold down
00:00:15:00.000

Related Information

Commands:

show

Books:

- [z/OS DCE Administration Guide](#)
show

Displays current information about the DCE DTS entity.

Format

dtscp sho[w] attribute-group | attribute-name

Parameters

attribute-group The name of an attribute group or individual attribute to be displayed. The following values are valid:
   • all
   • all characteristics
   • all counters
   • all status
   • global servers
   • local servers

attribute-name The name of a specific attribute from the characteristics, counters, or status group. The attribute specifiers global servers and local servers do not contain any other attributes.

Usage

The show command displays the names and values of a specified attribute or attribute group. For attribute groups, if you do not supply a group name with the all argument, all characteristics and their values are displayed. The following sections describe each of the values for the individual attributes associated with the characteristics, counters, and status attribute groups.

Note: The attributes displayed by the show command are different, depending on whether you request information about a server or a clerk.

Characteristics: Characteristic arguments have an assigned default and a value range. Character arguments can contain a maximum of 80 characters and are recalculated to a normalized date format. For example:


Note: The rule for a three-character abbreviation (as discussed in the description for “DTS Control Program” on page 615) holds true for all of these commands.

acting [courier [role]] Specifies whether a backup courier is currently functioning as a courier. If the role is noncourier, the node is not attempting to synchronize with global servers.
   Default: noncourier
   Value: courier or noncourier

automatic [tdf [change]] Specifies whether automatic changes to the time differential factor are enabled or disabled; the value is determined by the operating system.
   Default: false
   Value: true/false
show

check [interval]  Specifies the amount of time between checks for faulty servers. Applicable only to servers that have external time providers. This characteristic is shown only for servers.
Default: 0-01:30:00.00
Value: 0-00:00:30.000 - 10675199-02:48:05.478
The value range is from 30 seconds to infinity.

clock adjustment [rate]
Specifies the rate at which the DTS server or clerk entity adjusts the node’s clock during a synchronization.
Default: 10 000 000 nsec/sec

clock resolution
Specifies the amount of time between system clock ticks. The value is determined by the operating system.
Default: 10 000 000 nsec
Value: Determined by the operating system.

courier [role]
Specifies a server’s interaction with the set of global servers. This characteristic is shown only for servers.
Default: noncourier
Value:
backup courier  The local server becomes a courier if none is available on the LAN.
courier  The local server synchronizes with the global set of servers.
noncourier  The local server does not synchronize with the global set of servers.

DTS [version]
Specifies the DCE DTS software version installed on the node.

epoch number
Specifies the server’s epoch number. The change command modifies this characteristic. This characteristic is shown only for servers.
Default: 0
Value: 0-255

error [tolerance]
Specifies the maximum separation allowed between the local clock and the computed time before synchronizations become abrupt rather than gradual (monotonic).
Default: 0-00:10:00.000
Value: 0-00:00:00.500 - 10675199-02:48:05.478

global set [timeout]
Specifies the amount of time the node waits for a response to a global server synchronization request before sending another request or declaring a global server to be unavailable. The node sends out a number of query-attempts requests before declaring a server unavailable that is not responding. The number of query attempts is controlled by the query-attempts characteristic. The default number of query attempts is three.
Default: 0-00:01:30.000
Value: 0-00:00:00.000 - 0-00:10:00.000

local set [timeout]
Specifies the amount of time the node waits for a response to a local server synchronization request before sending another request or declaring a server to be unavailable. The node sends out a number of query-attempt requests before declaring
show a server unavailable that is not responding. The number of query attempts is controlled by the query-attempts characteristic. The default number of query attempts is three.

Default: 0-00:01:00.000
Value: 0-00:00:00.000 - 0-00:02:00.000

**local time differential factor**

Specifies the Time Differential Factor (TDF), which is the amount of time the server varies from Greenwich mean time (GMT) or Coordinated Universal Time (UTC). This parameter gets updated after the clock is adjusted.

Default: 0-00:00:00.000
Value: -13-00:00:00 - 13-00:00:00

**maximum clock [drift [rate]]**

Specifies the worst-case drift rate of the node’s clock, in nanoseconds per second, as determined by the manufacturer’s specifications.

Default: 1 000 000 nsec/sec

**maximum inaccuracy**

Specifies the inaccuracy limit for the node. When the node exceeds the maximum inaccuracy setting, it attempts to synchronize.

Default: 0-00:00:00.100
Value: 0-00:00:00.0 - 1-00:00:00.0

**next [tdf [change]]**

Specifies the future time when the time differential factor is automatically changed. The value is determined by the operating system.

Default: See description.
Value: Determined by the operating system.

**query [attempts]**

Specifies the number of attempts that a node makes to contact a server before the node considers the server unavailable.

Default: 3
Value: 1-10

**server entry [name]**

Specifies a server’s ACL entry name; `<hostname>` represents the name of the system or node that is the server’s client. The default setting is the recommended value. This characteristic is shown only for servers.

Default: hosts/<hostname>/dts-entity
Value: The default setting.

**server group [name]**

Specifies the security group name for the time servers within the cell. The default setting is the recommended value.

Default: subsys/dce/dts-servers
Value: The default setting.
server principal [name]
Specifies a server’s principal name for authentication purposes; <hostname> represents the name of the system or node that is the server’s client. The default setting is the recommended value. This characteristic is shown only for servers.
Default: hosts/<hostname>/self
Value: The default setting.

servers required
Specifies the minimum number of servers required for a synchronization. Settings of 1 or 2 may cause unreliable computed times.
Default: 3
Value: 1-10

synchronization hold [down]
Specifies the interval a node must wait to synchronize.
Clerks:
Default: 0-00:10:00.0
Value: 0-00:00:30.0 - 01-00:00:00.00
Servers:
Default: 0-00:02:00.0
Value: 0-00:00:30.0 - 01-00:00:00.00

time provider present
Specifies whether the entity used an external time-provider at the last successful synchronization. This attribute applies to servers only.

time representation version
Specifies the timestamp format used by the node.

type
Specifies whether the node is a DTS server or clerk. The create command modifies this characteristic.

Counters:
clock settings
Specifies the number of times the node clock has been set abruptly (not monotonically).

creation [time]
Specifies the time at which the DTS entity was created and the counters were initialized.

different [epochs [detected]]
Specifies the number of times the node received time-response messages from servers or clerks that had epoch numbers different from its own. This counter is shown only for servers.

disable [directives [completed]]
Specifies the number of times the DCE DTS entity has been disabled.

enable [directives [completed]]
Specifies the number of times the DCE DTS entity has been enabled.

epoch changes [completed]
Specifies the number of times the server’s epoch has changed.

insufficient [resources [detected]]
Specifies the number of times the node could not allocate virtual memory.
local servers not in group
Specifies the number of times that a local server was contacted, but was not in the DTS security group.

local times not [intersecting]
Specifies the number of times the node's time interval failed to intersect with the computed interval of the servers.

no global servers detected
Specifies the number of times the courier server could not contact any global servers. This counter is shown only for servers.

server times not [intersecting]
Specifies the number of times a server has detected faulty servers (other than itself).

server not in group
Specifies the number of times that global servers selected for synchronization are not members of the time service group. This counter is shown only for servers.

servers not responding
Specifies the number of times the courier server could not contact a specific global server. This counter is shown only for servers.

synchronizations completed
Specifies the number of times the node successfully synchronized.

system errors detected
Specifies the number of times a DTS operation detected a system error.

time provider failures [detected]
Specifies the number of times the external time-provider signalled a failure or the node could not access the time-provider.

time provider timeouts [detected]
Specifies the number of times a dtsd server process initiated contact with a time-provider and did not receive the initial response within the interval specified by local set [timeout]. The default interval is five seconds. This counter is shown only for servers.

time representation mismatches [detected]
Specifies the number of times the local node failed to process a received message containing an incompatible timestamp format.

too few [servers [detected]]
Specifies the number of times a node failed to synchronize because it could not contact the required minimum number of servers.

updates [initiated]
Specifies the number of times a server has attempted to update its clock. This counter is shown only for servers.

Status:
current time
Specifies the current time on the node.
global servers
Specifies the set of global servers known by the node.
last synchronization
Specifies the computed time at the last attempted synchronization.
local servers
Specifies the set of local servers known by the node.
show

state  Specifies the state of the DCE DTS entity.

- **off**: The DCE DTS entity is disabled.
- **on**: The DCE DTS entity is enabled.
- **synchronizing**: The DCE DTS entity is synchronizing.
- **updating**: The DCE DTS entity is updating the time.

uid  Specifies the entity’s unique identifier, which is generated when the entity is created. This attribute is shown only for servers.

**Privilege Required:** You must have write permission on the ACL associated with the DTS entity to run the command.

**Note:** The `dtscp show` command is supported in z/OS DCE by the `dcecp dts show` command.

**Examples**

- The following command displays the current time:
  ```
  dtscp> show current time
  ```

- The following command displays all of the entity’s characteristic attribute settings:
  ```
  dtscp> show all
  Check Interval = +0-01:30:00.0001-----
  Epoch Number = 0
  Error Tolerance = +0-00:10:00.0001-----
  Local Time Differential Factor = -0-04:00:00.0001-----
  Maximum Inaccuracy = +0-00:00:00.1001-----
  Servers Required = 3
  Query Attempts = 3
  Local Set Timeout = +0-00:00:05.0001-----
  Global Set Timeout = +0-00:00:15.0001-----
  Synchronization Hold Down = +0-00:02:00.0001-----
  Type = Server
  Courier Role = NonCourier
  Acting Courier Role = NonCourier
  Clock Adjustment Rate = 400000000 nsec/sec
  Maximum Clock Drift Rate = 1000000 nsec/sec
  Clock Resolution = 10000000 nsec
  DTS Version = V1.0.1
  Time Representation Version = V1.0.0
  Time Provider Present = FALSE
  Automatic TDF Change = FALSE
  Next TDF Change = 1997-10-31-06:00:00.000+00:0010.000
  Server Principal Name = hosts/system1/self
  Server Entry Name = hosts/system1/dts-entity
  Server Group Name = subsys/dce/dts-servers
  ```

- The following command displays the current values of all characteristic attributes: This command produces the same output as the `show all` command.
  ```
  dtscp> show all characteristics
  ```

- The following command displays all of the local servers known to the entity:
dtscp> **show local servers**

**Known Servers**

```
Local /.*sisyphus.osf.org/hosts/system2/self
  Last Time Polled   = 1995-10-15-21:01:46.124+00:00:00:00:00:00
  Last Observed Time = 1995-10-15-21:03:09.041+00:00:00:00:00
  Last Observed Skew = +0:00:01:22.917---
  Used in Last Synchronization = TRUE
  Transport Type = RPC

Local /.*sisyphus.osf.org/hosts/system3/self
  Last Time Polled   = 1995-10-15-21:01:46.124+00:00:00:00:00:00
  Last Observed Time = 1995-10-15-21:01:46.143+00:00:00:00:00
  Last Observed Skew = +0:00:00:00.01911. 625
  Used in Last Synchronization = TRUE
  Transport Type = RPC
```

- The following command displays the current values of all counter attributes:

```
dtscp> **show all counters**
```

```
  Creation Time   = 1995-10-14-16:23:57.801+00:00:00:00:00:00
  Local Times Not Intersecting = 0
  Server Times Not Intersecting = 0
  Different Epochs Detected = 0
  Too Few Servers Detected = 0
  Time Provider Timeouts Detected = 1
  Protocol Mismatches Detected = 0
  Time Representation Mismatches Detected = 0
  No Global Servers Detected = 0
  Servers Not Responding = 0
  Clock Settings = 0
  Epoch Changes Completed = 0
  System Errors Detected = 0
  Synchronizations Completed = 865
  Updates Initiated = 0
  Enable Directives Completed = 1
  Disable Directives Completed = 0
  Insufficient Resources Detected = 0
  Time Provider Failures Detected = 0
  Local server not in group = 0
  Servers not in group = 0
```

- The following command displays the current values of all status attributes:

```
dtscp> **show all status**
```

```
  UID = 00004e34-5e1c-2c87-8500-0005a0d4582
  Last Synchronization = 1995-10-15-21:05:43.023+00:00:00:00
  State = On
```

- The following command displays the current value of the courier role attribute:

```
dtscp> **show courier role**
```

```
  Courier Role = NonCourier
```

- The following command displays the server entry name for this server:

```
dtscp> **show server entry name**
```

```
  Server Entry Name = hosts/system1/dts-entity
```

- The following command displays the current state of the DTS entity:
show

dtscp> **show state**
State = On

- The following command displays the current value of the check interval attribute:

dtscp> **show check interval**
Check Interval = +0-01:30:00.0001-----

- The following command displays the current value of the servers times not intersecting counter:

dtscp> **show servers times not intersecting**
Server Times Not Intersecting = 0

**Related Information**

Commands:

set
synchronize

Causes the DTS entity to synchronize the clock in the system where the command is entered.

Format

dtscp syn[chronize] [set clock boolean]

Parameters

set [clock] boolean

Specifies whether the clock is abruptly set or gradually adjusted to the computed time. This argument is optional. The following values are valid:

<table>
<thead>
<tr>
<th>boolean</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>The clock is gradually adjusted. This is the default condition.</td>
</tr>
<tr>
<td>true</td>
<td>The clock is abruptly set.</td>
</tr>
</tbody>
</table>

Usage

The synchronize command causes the DTS clerk or server to solicit time intervals from servers, compute the intersection of the time intervals, and adjust the system clock to the midpoint of the computed time interval. This command overrides the functions of the synchronization hold down characteristic.

Privilege Required: You must have write permission on the ACL associated with the DTS entity to run the command.

Notes:

1. The synchronize command does not run if the entity is already synchronizing or is disabled; the entity must be enabled.

2. The synchronize command is supported in z/OS DCE by the dcecp dts synchronize command.

Examples

1. This example shows how to start a synchronization for the entity, followed by a gradual clock adjustment:
   
dtscp> synchronize

2. This example shows how to start a synchronization for the entity, followed by an abrupt reset of the clock:
   
dtscp> synchronize set clock true
unadvertise

Removes the global server entry from the cell profile.

Format

dtscp una[dvertise]

Usage

The unadvertise command causes DTS to remove the server's name from the cell profile and binding from the related CDS entry, deleting the server's global status.

Privilege Required: You must have write permission on the ACL associated with the DTS entity to run the command.

Note: The unadvertise command is supported in z/OS DCE by the dcecp dts configure command.

Examples

dtscp> unadvertise

Related Information

Commands:

advertise
**update**
Gradually adjusts the clock on the local node to the time specified by the argument.

**Format**

```
dtscp update time absolute-time
```

**Parameters**

- `time absolute-time`
  Specifies the absolute time to which the clock is adjusted. This argument is required.

**Usage**

The `update` command gradually adjusts the system clock to a new time. The difference between the current clock value and the absolute time specified in the argument is used to adjust the clock.

**Privilege Required:** You must have write permission to run the command.

**Notes:**

1. The `update` command is valid only for servers. The combined time and inaccuracy value you specify must be contained within the interval formed by the current time and inaccuracy. That is, the new setting must be more accurate than the current time.
2. The `update` command is supported in z/OS DCE by the `dcecp clock set` command.

**Examples**

This example shows how to update the time for a server; the clock is gradually adjusted to the specified time.

```
dtscp> update time 1996-12-30-11:24:00.000-05:00I0.000
```

**Related Information**

Commands:

- change
update
Chapter 6. Security Service

This chapter introduces you to the DCE Security Service (SEC), and describes the SEC administrative facilities and their associated commands. The SEC administrative facilities include:

- The access control list editor, `acledit`, which allows you to modify an object’s ACLs.
- The `dcelogin` command, which allows a user to login to DCE and obtain a ticket to the security server daemon.
- The `kdestroy`, `klist`, and `kinit` commands, which operate on a user’s ticket cache.
- The registry editor, `rgyedit`, which allows you to modify the security registry database.
- The `secadmin` command that provides a registry replica administration tool.
- The `sec_create_db` command, which creates the default registry database when a DCE cell is reconfigured.
- The `sec_export_db` command, which lets you create a portable version of the security registry database.
- The `sec_import_db` command, which lets you merge an exported registry into the security registry database.
- The security server daemon `secd`.
- The `loginact` command, which allows the DCE administrator to query or reset login activity information for a DCE user.

All the DCE Security administrative facilities can either be started as a UNIX System Services command, a TSO/E subcommand, or by batch JCL.

The following chapter describes each of the DCE Security facilities listed above. For samples of batch JCLs, refer to the `z/OS DCE Administration Guide`.

The following Security Service commands and daemons are not supported in z/OS DCE, and are not described in this chapter:

- `chpass`
- `passwd_export`
- `passwd_import`
- `passwd_override`
- `sec_salvage_db`

Some SEC commands are supported by the `dcecp` command. The following table shows the `dcecp` commands that support these SEC commands:

<table>
<thead>
<tr>
<th>SEC Command</th>
<th>dcecp Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>acledit</td>
<td>acl</td>
</tr>
<tr>
<td>dcelogin</td>
<td>login</td>
</tr>
</tbody>
</table>
Table 16 (Page 2 of 2). DCECP Equivalent Commands for Security Commands

<table>
<thead>
<tr>
<th>SEC Command</th>
<th>dcecp Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>rgidget</td>
<td>account</td>
</tr>
<tr>
<td></td>
<td>group</td>
</tr>
<tr>
<td></td>
<td>organization</td>
</tr>
<tr>
<td></td>
<td>principal</td>
</tr>
<tr>
<td></td>
<td>registry</td>
</tr>
<tr>
<td>secadmin</td>
<td>registry</td>
</tr>
</tbody>
</table>
**acledit**

Edits or lists an object's ACLs.

**Format**

```
acledit [-e] pathname / -addr string_binding component_name} [-ic | -io] 
[ command_line_options] [-v]
```

---

**Note to Users**

In the shell environment, the `acledit` command is entered as `acl_edit`.

**Options**

- **-e**
  Specifies that the ACL on the Name Service entry is to be edited. The `-e` option must be used with the `pathname` argument.

  The `-e` option is especially useful in case of an ambiguous `pathname`. If the `pathname` specified in the `acledit` command line is the name of a leaf object in the Name Service (that is, if it is not the name of a directory), it can be interpreted in one of two ways: it can refer to the Name Service entry itself, or it can refer to the object that is referenced by that Name Service entry. When such a `pathname` is specified, the `-e` option directs `acledit` to the ACL on the Name Service entry. However, the `-e` option can always be used, provided only that `pathname` specifies a leaf object in the Name Service.

- **-addr string_binding component_name**
  The `-addr` option lets you identify the object whose ACLs you want to edit, by supplying the RPC binding handle of the ACL Manager that controls access to the object (with the `string_binding` argument), and by supplying the relative path name of the object (with the `component_name` argument). Because you have identified the RPC binding handle, you can specify only the object's name for `component_name`. The most common way to identify the object whose ACLs you want to manipulate is through the `pathname` argument, described below. The `-addr` option is used primarily by applications that do not use the Naming Service, but do use the generic ACL Manager. It can also be used if the Naming Service is unavailable.

- **-ic**
  For container objects only, specifies that the object's Initial Container Creation ACL is to be edited. The Initial Container Creation ACL is applied by default to any containers created within the container for which the Initial Container Creation ACL is defined. If this option is specified and the object named in `pathname` is not a container, an error is returned.

- **-io**
  For container objects only, specifies that the object's Initial Object Creation ACL is to be edited. The Initial Object Creation ACL is applied by default to any simple objects (objects that are not containers) created within the container for which the Initial Object Creation ACL is defined. If this option is specified and the object is not a container, an error is returned.

- **-v**
  Run in verbose mode.

- **-stdin**
  Used in z/OS DCE if the PARM field in the JCL exceeds 100 characters. When this option is used, the arguments to `acledit` must be entered through a SYSIN statement in the JCL.
acledit

Parameters

pathname

The full path name of the object whose ACL is to be viewed or edited. If the object
is in another cell, pathname must be fully qualified to include the cell identifier. The
cell you specify must be running an ACL manager.

command_line_options

The command-line options that act on the object specified by pathname are entered
as part of the command string that starts acledit. Only one command-line option
can be specified per invocation. The options appear in the following list. For a
more detailed description of the command functions, see "Interactive
Subcommands" on page 644.

-\textbf{m \textit{acl\_entry}} \hspace{1em} Adds a new ACL entry or changes the permissions of an
existing entry. You can enter multiple entries, each separated
by a space.

-\textbf{d \textit{acl\_entry}} \hspace{1em} Deletes an existing entry from the ACL associated with the
specified object. You can enter multiple entries, each
separated by a space.

-\textbf{s \textit{acl\_entry}} \hspace{1em} Replaces the ACL information associated with this object with
\textit{acl\_entry}. All existing entries are removed and replaced by
the newly specified entries. If you specify the -s
subcommand, you cannot specify the -f or -k subcommand.
You can enter multiple entries, each separated by a space.

-\textbf{f DD:\textit{file}} \hspace{1em} Assigns the ACL information contained in \textit{file} to the object.
All existing entries are removed and replaced by the entries in
the file. If you specify the -f subcommand, you cannot specify
the -s or -k subcommand.

-\textbf{k} \hspace{1em} Removes all entries, except entries of type \textit{user\_obj} (if they
are present). If you specify the -k subcommand, you cannot specify
the -f or -s subcommand.

-\textbf{l} \hspace{1em} Lists the entries in the object’s ACL.

The command-line options are evaluated in the following order:

1. -s or -f or -k
2. -d
3. -m
4. -l

Note: With the exception of the following subcommands, this command is supported by the \texttt{dcecp acl}
list and \texttt{acl modify} commands.

- abort
- commit
- exit
- help
- test access
Usage

acledit is a client agent program that, when started, binds to the specified object's ACL manager (which is implemented in the object's server). You can manipulate the object's ACL through the standard DCE-wide ACL interface. This interface is the same one used by the sec_acl_...() routines documented in the [z/OS DCE Application Development Reference](https://www.ibm.com).

The object whose ACL is to be operated on is specified by the pathname argument; acledit must be able to bind to this object (or, to be precise, to the server that maintains the object) by performing an import operation from the specified Name Service entry. Also, the binding information that acledit imports from the entry must contain an object UUID that uniquely identifies the desired object to its server when acledit’s remote procedure calls are received by the server’s ACL manager. Servers do not explicitly export the ACL interface; acledit imports the binding for the specified object by specifying the NULL interface. In this way, the name space object entries can be used to export both the objects themselves and the ACLs of the objects, thus conserving name-space space.

For an application to make its ACL accessible through the standard DCE ACL interface, and usable by the acledit command and the sec_acl_...() library interface routines, it must export separate bindings in separate entries for the ACL in which the objects are defined. This can be done directly in the name space, or by means of a name space junction. For further information, see the Representation of Objects with ACLs in the Namespace in the [z/OS DCE Application Development Guide: Core Components](https://www.ibm.com).  

Note:  
The acledit command is supported in z/OS DCE by the dcecp acl command.

All acledit subcommands act on the object specified by pathname when you start acledit. You can start acledit in either command-line or interactive mode:

- To start command-line mode, enter the command, the object's path name, options, and the command-line subcommand on the line that starts acledit. Only one command-line subcommand can be entered for each acledit invocation.

- To start interactive mode, enter only acledit, the object's path name, and options. The sec_acl_edit prompt is displayed after you start acledit. In this mode, you enter interactive subcommands that let you edit and view entries in the object's ACL and view help information about the acledit command itself.

Changes you make in command-line mode are saved when you enter the command. In interactive mode, you must explicitly save your changes. To do so, use the commit subcommand to save the changes without exiting acledit, or the exit subcommand to save the changes and exit acledit. Use the abort subcommand to exit acledit and save none of the changes that you have made.

Note: When you start acledit for a specific object's ACL, that ACL is not locked. This means that it is possible for multiple users to edit the ACL simultaneously, with each change overwriting the previous changes. For this reason, the number of users assigned rights to change a particular ACL should be tightly controlled and limited to one user, if possible.

Enter all non-interactive acledit commands from native TSO/E or the shell only. If these commands and their arguments are entered from the ISPF command line they are converted to lowercase characters. This is a problem for arguments that have mixed or all uppercase characters.

If you run acledit in batch mode, the input to the control programs and the output for each command are written to the job execution log. You can then determine which input commands were successful and which were not.
Interactive Subcommands

The following subcommands are available when acledit is started in interactive mode. They all act on the ACL associated with the object specified by pathname when acledit was started.

? Displays the available acledit subcommands.

ab[ort] Exits acledit without saving the changes to the object's ACL.

as[ign] DD:filename
Applies the ACL entries in filename to the specified object. This subcommand removes existing entries and replaces them with the entries in the file.

c[ell] name
Sets the default cell for the ACL. This subcommand is used primarily to facilitate the copying of ACLs to different cells. The default cell name is not changed until the subcommand is run again specifying a different cell. The name supplied will be interpreted as a name in the local cell unless the name begins with the global prefix /... Any over-qualification of the cell name is ignored. For instance, if you specify the name of a principal using the cell subcommand, only the part of the name that specifies the cell is used, and the part that specifies the principal within the cell is ignored.

c[o][mmit] Saves all changes made to the ACL without exiting acledit.

d[elete] acl_entry Deletes the specified ACL entry.

e[xit] Exits from acledit and saves the changes to the object's ACL.

g[et_access]
Displays the permissions granted in the specified object's ACL to the principal that started acledit.

h[elp] [command...]
Initiates the help facility. If you enter only the command help, acledit displays a list of all commands and their functions. If you enter help and a command (or commands separated by a space), acledit displays help information on the specified commands. Entering help sec_acl_entry displays information about ACL entries.

k[ill_entries] Removes all ACL entries except the user_obj entry, if it exists.

l[ist] Lists the entries in the object's ACL.

m[odify] acl_entry Adds a new ACL entry or replaces an existing ACL entry. This subcommand affects a single ACL entry. To add or replace all of an object's ACL entries, see the su[bstitute] subcommand.

p[ermissions] Lists the available permission tokens and explanations.

s[u][bstitute] acl_entry [acl_entry ...]
Replaces all ACL entries with the entry or entries specified. This subcommand removes all existing entries and adds the ones specified by acl_entry. To replace only a single ACL entry, see the m[odify] subcommand.

t[est_access] [ permissions.... ]
Tests whether the permissions specified in the command are granted to the principal under whose DCE identity the acledit command was started. The option returns Granted if the permissions are granted, or Denied if they are not.
ACL Entries

An ACL entry has the following syntax:

type[:key]:permissions

Where:

type Identifies the role of the ACL entry.
key Identifies the specific principal or group to whom the entry applies. For an entry type of extended, key contains the ACL data.
permissions The ACL permissions.

A description of each syntax component follows:

Type: The type tag identifies the role of the ACL entry. Valid types are:

- user_obj Permissions for the object’s real or effective user.
- group_obj Permissions for the object’s real or effective group.
- other_obj Permissions for others in the local cell who are not otherwise named by user and group entries.
- user Permissions for a specific principal user in the local cell. This type must include a key that identifies the specific principal.
- group Permissions for a specific group in the local cell. This type must include a key that identifies the specific group.
- foreign_user Permissions for a specific, authenticated user in a foreign cell. This type must include a key that identifies the specific principal and the principal’s cell.
- foreign_group Permissions for a specific, authenticated group in a foreign cell. This type must include a key that identifies the specific group and the group’s cell.
- foreign_other Permissions for all authenticated principals in a specific foreign cell, unless those principals are specifically named in an ACL entry of type foreign_user or members in a group named in an entry of type foreign_group. This type must include a key that identifies the specific foreign cell.
- any_other Permissions for all authenticated principals unless those principals match entries in the ACL.
- mask_obj Permissions for the object mask that is applied to all entry types except user_obj, other_obj, and unauthenticated.
- user_obj_delegate Permissions for an intermediary acting for the object's real or effective user.
- group_obj_delegate Permissions for an intermediary acting for members of the object's real or effective group.
- other_obj_delegate Permissions for an intermediary acting for all other principals in the default cell, unless they are specifically named in ACLs of an entry type of user, are members of a group named in an ACL with an entry type of group, or match the principal indicated by the user_obj or group_obj entry.
- user_delegate Permissions for an intermediary acting for a specific principal in the default cell of the ACL. This ACL entry type requires a key that is a principal name.
acledit

**foreign_user_delegate**
Permissions for an intermediary acting for a specific principal in a foreign cell, one other than the default cell of the ACL. You must identify the principal by supplying a principal name and cell name as a key.

**foreign_group_delegate**
Permissions for an intermediary acting for a specific group in a foreign cell, one other than the default cell of the ACL. You must identify the group by supplying a group name and a cell name as a key.

**foreign_other_delegate**
Permissions for an intermediary acting for other principals in a specific foreign cell, one other than the default cell of the ACL, that are not specifically named in the ACL entries of entry type `foreign_user` or are members of a group named in an ACL entry of type `foreign_group`. You must identify the foreign cell by supplying a cell name as a key.

**any_other_delegate**
Permissions for an intermediary acting for all other principals in local or foreign cells unless they match a more specific entry in the ACL.

**unauthenticated**
Maximum permissions applied when the accessor does not pass authentication procedures. This entry is used for principals who have failed authentication because of keys that are not good, principals who are entirely outside of any authentication cell, and principals who choose not to use authenticated access. Permissions granted to an unauthenticated principal are masked with this entry, if it exists. If this entry does not exist, access to unauthenticated principals is always denied.

**extended**
A special entry that allows client applications running at earlier DCE revisions to copy ACLs to and from ACL managers running at the current DCE version without losing any data. The `extended` entry allows the application running at the earlier version to obtain a printable form of the ACL. The extended ACL entry has the following form:

```
uuid.ndr.ndr.ndr.ndr.number_of_bytes.data
```

- **uuid**
  Identifies the type of extended ACL entry. (This UUID can identify one of the ACL entry types described here, or an undefined ACL entry type.)

- **ndr.ndr.ndr.ndr**
  Up to four Network Data Representation (NDR) format labels (separated by periods) that identify the encoding of data.

- **number_of_bytes**
  A decimal number that specifies the total number of bytes in `data`.

- **data**
  The ACL data in hexadecimal form. (Each byte of ACL data is 2 hexadecimal digits.) The ACL data includes all of the ACL entry specifications except the permissions that are entered separately. The `data` is not interpreted; the ACL manager being passed the data understands it.

**Key:** The key identifier (principal or group name) of the principal or group to which the ACL entry applies. For entries of entry type `extended`, key is the data passed from one ACL manager to another. A key is required for the following types of ACL entries:

- **user**
  Requires a principal name only.

- **group**
  Requires a group name only.

- **foreign_user**
  Requires a fully qualified cell name in addition to the principal name.

- **foreign_group**
  Requires a fully qualified cell name in addition to the group name.
foreign_other  Requires a fully qualified cell name.

Permissions:  Permissions is the set of permissions that defines the access rights conferred by the entry. Because each ACL manager defines the permission tokens and meanings appropriate for the objects it controls, the actual tokens and their meanings vary. For example, the File System, the Naming Server, and the Registry Service each implement a separate ACL manager, and each can use a different set of tokens and permissions. File system objects, objects in the name space, and registry objects can each use different permissions. Use the permissions subcommand to display the currently available tokens and their meanings. See the RPC, CDS, and SEC chapters to obtain a more detailed description of their specific permissions.

Examples

1. This example uses the interactive interface to set permissions for the unauthenticated and object mask entry type:
   
   ```
   sec_acl_edit> m mask_obj:rwx
   sec_acl_edit> m unauthenticated:r
   ```

2. This example uses the interactive interface to set permissions for the effective user, group, and others in the ACL’s cell:
   
   ```
   sec_acl_edit> m user_obj:crwx
   sec_acl_edit> m group_obj:rwx
   sec_acl_edit> m other_obj:rwx
   ```

3. This example uses the command-line interface to start acledit and assign permissions for the file progress_chart to the authenticated user mike in the local cell:
   
   ```
   sec_acl_edit /.../dresden.com/afs/walden/progress_chart -m user:mike:crwx
   ```

   Because this entry will be filtered through the object mask (mask_obj), which specifies only rwx permissions, the actual permissions will be rwx, not crwx. The list subcommand will show those permissions as:
   
   ```
   user:mike:crwx #effective -rwx---
   ```

4. This example uses the interactive interface to set permissions for the authenticated foreign user named burati in the cell named /.../usc-cs.uscal.edu:
   
   ```
   sec_acl_edit> m foreign_user:/.../usc-cs.uscal.edu/sailing/staff/burati:rwx
   ```

5. This example uses the noninteractive command-line interface to start acledit and set the Initial Container Creation permissions for the directory named walden:
   
   ```
   sec_acl_edit /.../dresden.com/afs/walden -ic -m /user:walden:crwxid
   ```
dcelogin

Validates a principal’s identity and obtains the principal’s network credentials.

Format

dcelogin [principal_name] [password] [-k {keyfile} | -c] [-r]

- **Note to Users**
  Use `dcelogin` to log in to z/OS DCE in batch or TSO/E, and `dce_login` to log in to z/OS DCE in the shell.

Options

Options

- `-k {keyfile} | -c` Use `-k {keyfile}` to specify the keyfile that contains the account password for the specified principal.
  Use `-c` to specify that the identify is to be certified.
- `[-r]` Specifies that the current login identity is to be refreshed.

Usage

The `dcelogin` command obtains network credentials for a DCE principal. You can also use the `dcelogin` command for DCE configuration. The `dcelogin` command validates a principal’s identity and obtains the principal’s network credentials. If the `-c` option is supplied, the command also certifies the principal’s identity. If the `-c` option is not specified, the principal’s identity will automatically be certified by the first authenticated RPC call (for example, starting `rgyedit`).

The `principal_name` argument specifies the name of the principal who is logging on. The `password` argument specifies the principal’s password. If you do not supply a principal name or a principal password, `dcelogin` prompts for them. If you enter them both on the command line, you must specify the principal name followed by the password.

When issuing the `dcelogin` command from the TSO/E environment, the login will fail if the password in the command ends with a dash (`-`), because the dash is treated as a line continuation character.

If you use a password that ends with a dash, the command will process correctly if:

- An additional space follows the password on the command line, or
- The password is not entered on the command line, then the `dcelogin` program prompts the user for the password

Notes:

1. If the password contains one or more blanks, do not specify the password on the command. Instead, wait for the prompt.

2. If the clocks on the Security server and client machines are not synchronized to within 2 or 3 minutes of each other, you may receive a password validation error. For information on how to hide your password when you log in, refer to the [z/OS DCE Administration Guide](#).
3. If you are enabled for DCE automatic sign-on and are not currently logged in to DCE, DCE will log you
in before the command is processed. If you do not want to be automatically logged on to DCE, put
_EUV_AUTOLOG=NO in your ENVAR file before issuing the command.

Files
A successful dcelogin results in a credentials cache file being created in the directory
/opt/dcelocal/var/security/creds. The naming syntax for this file is dcecred_nnnnnnnnnn (where nnnnnnnn
is a timestamp).

The dcelogin command sets the KRB5CCNAME environment variable to reference the credentials cache
file. The declaration of this variable is contained in a separate file. The default file, krb5ccname, is in the
user's home directory. You can override the default file in one of the following ways:

- By using DDNAME EUVSKRB5
- By using the environment variable _EUV_SEC_KRB5CCNAME_FILE.

If a user's account expires, the administrator can use the loginact command to re-enable the user
account.

Examples

The following is an example of how to use multiple user IDs within one TSO/E session:

1. Enter dcelogin and log in as a normal DCE user with no administrative privileges. A
   home_directory/krb5ccname file is created. This is the Credentials Cache Name file that will
   reference the ticket cache. Any DCE command started, such as rgyedit, acledit, klist, and cdscp,
   will obtain the value of the KRB5CCNAME environment variable from the Credentials Cache Name
   file.

2. To perform administrative tasks start tso setkrb5 home_directory/krb5ccname/celladmn. A new
   Credentials Cache Name file is created.

3. Perform another dcelogin as the cell administrator principal. The
   home_directory/krb5ccname/celladmn file is updated with a reference to the newly created ticket
   cache.

4. Perform the necessary administrative tasks.

5. Start tso setkrb5 home_directory/krb5ccname to reset the environment to its original,
   non-administrative dcelogin context.

At any time that the tickets are still valid, the environment can be switched to that of cell administrator or
the cell administrator can be switched to that of the environment by issuing the tso setenv
home_directory/krb5ccname/celladmn command. Another dcelogin is not required when you are
switching between environments as long as the tickets are still valid.

Related Information

Commands:

loginact

Books:

- z/OS DCE Administration Guide
- z/OS DCE User's Guide
kdestroy

kdestroy

Destroys a Kerberos credentials cache.

Format

kdestroy [-c cache_name] [-e time_delta]

Options

- `-c cache_name` Specifies the name of the credentials cache to be destroyed. The default credentials cache is destroyed if no command options are specified. This option is mutually exclusive with the `-e` option.

- `-e time_delta` Specifies that all credentials cache files containing expired tickets are deleted if the tickets were expired at least as long as the `time_delta` value.

Usage

The `kdestroy` command destroys a Kerberos credentials cache file. It supports the DCE and the Kerberos environments. The DCEKERN started task must be running on the local system to provide support for the software clock.

If the credentials cache contains a ticket to the `dce-rgy` service, the `kdestroy` command assumes that the DCE credentials should be destroyed as well as the Kerberos credentials. In this case, the DCE login functions purge the current DCE login context.

DCE supports inheriting the login context of the machine principal if a root (uid 0) user does not have a current login context. The `kdestroy` command will not destroy the login context of the machine principal if it was inherited. It will destroy the context if the machine principal credentials cache is explicitly specified by the `-c` option.

The `-e` option causes the `kdestroy` command to check all of the credentials cache files in the default cache directory (`/opt/dcelocal/var/security/creds`). Any file that contains only expired tickets that are expired for the time delta will be deleted. The time delta is expressed as `nwndnhmns` where:

- `n` represents a number
- `w` specifies weeks
- `d` specifies days
- `h` specifies hours
- `m` specifies minutes, and
- `s` specifies seconds

The components must be specified in this order, but any component may be omitted (for example, `4h5m` represents four hours and 5 minutes and `1w2h` represents 1 week and 2 hours). If only a number is specified, the default is hours.

In order to delete a credentials cache, the user must be the owner of the file or must be a root (uid 0) user.
Examples

To delete the default credentials cache, enter:

```
kdestroy
```

To delete all credentials cache with expired tickets older than 1 day, enter:

```
kdestroy -e 1d
```

Related Information

Commands:

```
klist
kinit
```
kinit

Obtains or renews the Kerberos ticket-granting ticket.

Format


Options

- **-r time** Specifies the renew time for a renewable ticket. The ticket may no longer be renewed after the expiration of this interval. The renew time must be greater than the end time. The ticket is not renewable if this option is not specified.

- **-R** Specifies that an existing ticket will be renewed. No other ticket options may be specified when renewing an existing ticket.

- **-p** Specifies that the ticket is to be proxiable.

- **-f** Specifies that the ticket is to be forwardable.

- **-l time** Specifies the ticket end time. The ticket may not be used after this interval expires unless it was renewed.

- **-c cache** Specifies the name of the credentials cache used by the **kinit** command. The default credentials cache is used if this option is not specified.

- **-k** Specifies that the key for the ticket principal is obtained from a key table. The user is prompted to enter that password for the ticket principal if this option is not specified.

- **-t keytab** Specifies the key table name. The default key table is used if this option is not specified and the -k option is specified. The -t option implies the -k option.

- **principal** Specifies the ticket principal. The principal is obtained from the credentials cache if the principal is not specified on the command line.

Usage

The **kinit** command obtains or renews a Kerberos ticket-granting ticket. It supports the DCE and the Kerberos ticket environments. The DCEKERN started task must be running on the local system to provide support for the software clock.

If the credentials cache contains a ticket to the **dce-rgy** service, the **kinit** command assumes that the DCE credentials should be refreshed. In this case, the principal name must be omitted or must be the same as the principal name obtained from the credentials cache. The DCE login functions will refresh the DCE login context. Any ticket options specified on the command line will be ignored.

If the credentials cache does not contain a ticket to the **dce-rgy** service, the **kinit** command assumes that the Kerberos credentials should be refreshed. If an existing ticket is not being renewed, the credentials cache will be re-initialized and will contain the new ticket-granting ticket received from the KDC. If the principal is not specified on the command line, it will be obtained from the credentials cache.

Ticket time values are expressed **nwndhnmns** where:

- **n** represents a number
- **w** specifies weeks
- **d** specifies days
• **h** specifies hours
• **m** specifies minutes, and
• **s** specifies seconds

The components must be specified in this order, but any component may be omitted (for example, 4h5m represents four hours and 5 minutes and 1w2h represents 1 week and 2 hours). If only a number is specified, the default is hours.

**Examples**

To obtain a ticket-granting ticket with a life time of 10 hours, which is renewable for 1 week, enter:

```bash
kinit -l 10h -r 1w my_principal
```

To renew an existing ticket, enter:

```bash
kinit -R
```

To refresh a DCE login context or to obtain a new ticket-granting ticket for the current principal, enter:

```bash
kinit
```

**Related Information**

Commands:

- klist
- kdestroy
klist

Displays the contents of a Kerberos credentials cache or key table.

Format


Options

-a
  Shows all tickets in the credentials cache, including an expired ticket. Expired
tickets are not listed if this option is not specified. This option is valid only when
listing a credentials cache.

-e
  Displays the encryption type for the session key and the ticket. This option is valid
only when listing a credentials cache.

-c
  Lists the tickets in a credentials cache. This is the default if nether the -c nor the -k
option is specified. This option is mutually exclusive with the -k option.

-f
  Shows the ticket flags using the following abbreviations:

  F  Forwardable ticket
  f  Forwarded ticket
  P  Proxiable ticket
  p  Proxy ticket
  D  Postdateable ticket
  d  Postdated ticket
  R  Renewable ticket
  I  Initial ticket
  i  invalid ticket
  H  Hardware preauthentication used
  A  Preauthentication used

  This option is valid only when listing a credentials cache.

-s
  Suppresses command output but sets the exit status to 0 if a valid ticket-granting
ticket is found in the credentials cache. This option is valid only when listing a
credentials cache.

-k
  Lists the entries in a key table. This option is mutually exclusive with the -c option.

-t
  Displays timestamps for key table entries. This option is valid only when listing a
key table.

-tk
  Displays the encryption key value for each key table entry. This option is valid only
when listing a key table.

filename
  Specifies the name of the credentials cache or key table. The default credentials
  cache or key table will be used if no filename is specified.

Usage

The klist command displays the contents of a Kerberos credentials cache or key table. It supports the
DCE and the Kerberos environments. The DCEKERN started task must be running on the local system in
order to provide support for the software clock.

If the credentials cache contains a ticket to the dce-rgy service, the klist command assumes that the DCE
credentials should be displayed as well as the Kerberos credentials. In this case, the DCE login functions
are used to obtain information about the current DCE login context. RPC requests to the DCE security server then converts DCE information into displayable text value.

Examples

To list all of the entries in the default credentials cache, enter:

```
klist
```

To list all of the entries in the `/krb5/my_keytab` key table with timestamps, enter:

```
klist -k -t /krb5/my_keytab
```

Related Information

Commands:

```
kinit kdestroy
```
**loginact**

Queries and resets the login activity information for a DCE principal.

---

**Note to Users**

In the shell environment, the `loginact` command is entered as `login_activity`.

---

**Format**

`loginact option [-p principal] [-n num_unsuccessful_attempts] [-d disable_time]`

**Options**

- `option` Represents the valid subcommand options. The options are:
  - `query` Displays the login activity information for the specified principal.
  - `reset` Resets the number of unsuccessful login attempts and the disable time attributes for the specified principal.
  - `help` Displays the `loginact` command usage statement.
  - `usage` Displays the `loginact` command usage statement.

- `-p principal` Specifies the principal name for which the login activity information is queried or reset. This information is required (and only valid for) the `query` and `reset` options.

- `-n num_unsuccessful_attempts` Specifies the number of unsuccessful login attempts for the principal supplied. This option is only valid with the `reset` command option.

- `-d disable_time` Specifies the time at which the principal is able to log into DCE. For example, the DCE administrator sets extended registry attributes (ERAs) for the principal that specify that the principal is allowed three unsuccessful login attempts before the account is disabled, and sets the disable time interval to 30 minutes (1800 seconds). When the principal unsuccessfully attempts to log into DCE for a third time, DCE resets the disable time to 30 minutes from the current system time. The principal is not able to log into DCE until this time or the disable time is reset by the DCE administrator.

  This option is only valid with the `reset` command option.

  The format of the `disable_time` option is `yyyy/mm/dd.HH:MM` as follows:

  - `yyyy/mm/dd` The year, month, and date until which the principal remains disabled.
  - `HH:MM` The local time in hours (0-23) and minutes (0-59) at which time the principal becomes enabled.

  There are special time values that can be used as operands for the `-d` option. These values are as follows:

  - `0` Sets the disable time to 0, immediately enabling the principal's account.
**Usage**

The `loginact` command displays or resets the login activity information for a specified principal.

The login activity information that is capable of being reset is defined in extended registry attributes (ERAs). These attributes allow the DCE administrator to control:

- The maximum number of unsuccessful login attempts allowed for a principal before disabling the DCE account associated with the principal, and
- The amount of time that the account remains disabled

Currently, attributes can only be set on a DCE principal. The `loginact` command allows the DCE administrator the capability of overriding these values by resetting the login activity information for a principal.

**Note:** Although this is intended for use as an administrator command, the `loginact` command can also be used by a DCE end user.

**Privilege Required:** The following permissions apply:

- `loginact reset` -- Requires `rm` permissions to the principal in the registry database
- `loginact query` -- Requires `r` permission to the principal in the registry database
- `loginact help` -- Requires no special permissions
- `loginact usage` -- Requires no special permissions

**Examples**

1. To query the login activity information for DCE principal `jonathan`, enter:

   ```
   loginact query -p jonathan
   ```

   Login activity information:
   - Principal name: jonathan
   - Current number of unsuccessful attempts: 0
   - Account disabled until: <not disabled>
   - Last successful login: Wed Sep 6 17:18:06 1996 from 9.130.79.38[1175]
   - Last unsuccessful login: <none> from <none>

   In this example, the `loginact query` command displays the login activity information for the principal `jonathan`. This information includes the current number of unsuccessful login attempts, the time until which the principal is disabled, and information regarding the last unsuccessful and successful login attempts.

2. To reset the disable time for the DCE principal `jordan`, enter:
loginact

`loginact reset -p jordan -d 96/10/16.08:52`

`loginact query -p jordan`

Login activity information:

Principal name: jordan
Current number of unsuccessful attempts: 1
Account disabled until: Mon Oct 16 08:52:00 1996
Last successful login: Mon Oct 9 09:32:29 1996 from 9.130.79.38[1141]

In this example, the `loginact reset` command changes the time until which the DCE principal `jordan` remains disabled to 8:52am on October 16, 1996.

Related Information

Commands:

`dcelogin`  `dcecp principal`  `dcecp xattrschema`
mvsexpt

Cross links RACF user IDs with their corresponding z/OS DCE principal IDs, to allow RACF interoperability and single sign-on.

Format

```
mvsexpt {-p1 [-e | -m [filename] | -u [filename]]}
    | -p2 [-r]}
```

Options

- **-p1**
  Specifies pass one, which creates an output file for pass two. Pass one uses the DCE Registry as its default input source. All principals in the Registry are used. Pass one creates the file, `/opt/dcelocal/var/security/adm/RACFWORK`, which contains RACF commands that create and update the RACF DCE segment for the DCE principal.

- **-e**
  Specifies that the Processed Entries File, `/opt/dcelocal/var/security/adm/PROCENTR`, is to be used instead of the DCE Registry as input to pass one.

- **-m**
  Specifies that the Identity Mapping File, `/opt/dcelocal/var/security/adm/ASUIDMAP`, is to be used instead of the DCE Registry as input to pass one.

- **-u**
  Specifies that the Principal Mapping File, `/opt/dcelocal/var/security/adm/PRNIDMAP`, is to be used instead of the DCE Registry as input to pass one.

- **filename**
  User-provided HFS file name. This is the name of an alternate input file for pass one.

- **-p2**
  Specifies pass two, which takes a file created by pass one and uses it to create or update the RACF user database.

  The RACF Work File, `/opt/dcelocal/var/security/adm/RACFWORK`, is the input file for pass two. The file contains RACF `adduser`, `altuser`, and `rdefine` commands.

- **-r**
  This option runs pass two with a corrected RACF Error File, `/opt/dcelocal/var/security/adm/RACFERS`.

Usage

1. This utility populates the RACF database with information from the DCE registry and enables single sign-on for DCE users. It is used for cross linking information between RACF user IDs and the corresponding DCE principal IDs. It enables single sign-on for DCE users. Use `mvsexpt` when starting the RACF database.

2. Before running the `mvsexpt` command, you must customize the file `/opt/dcelocal/etc/EXPTVAR`. See the [z/OS DCE Administration Guide](https://www.ibm.com/support/knowledgecenter) for a description of the files used by the `mvsexpt` command.

3. `mvsexpt` must be run from the z/OS UNIX System Services shell.

Examples

1. For pass one:
   ```
   mvsexpt -p1 -e
   ```
   creates the output file for pass two, using the Processed Entries file as input.

2. For pass one:
mvsexpt

mvsexpt -p1 -u /opt/dcelocal/var/security/adm/MYUPFILE
creates the output file for pass two, using a Principal Mapping File named by the administrator.

3. For pass two:
    mvsexpt -p2
    uses the command file created by pass one and processes those commands to the RACF database.

Related Information: Commands:
    mvsimpt
    storepw

Books:
    z/OS DCE Administration Guide
mvsimpt

Cross links RACF users IDs with their corresponding z/OS DCE user IDs, to allow RACF interoperability and single sign-on.

Format

mvsimpt {-p1 [-s filename] -pw mypwd} [-p2 [-r]}

Options

-p1  Specifies pass one, which creates an output file for pass two. Pass one uses a file generated by the RACF DB Unload Utility, IRRDBU00. The file is sorted for users with RACF DCE segments. Pass one creates the file /opt/dcelocal/var/security/adm/DCEWORK with the dcecp commands necessary to create DCE principals.

The default file name used by pass one is /opt/dcelocal/var/security/adm/RACFUNLD. This file is used if nothing is specified between -p1 and -pw.

-s  Specifies that a file other than the sorted IRRDBU00 output file is to be used as the DCE segments input file.

filename  User-provided HFS file name. This is the alternate DCE segments input file.

-pw  Specifies that a password follows. Required for pass one.

mypwd  Password of the DCE cell administrator executing the command.

-p2  Specifies pass two, which takes the command created by pass one and processes those commands to create new DCE principals.

The DCE Work File, /opt/dcelocal/var/security/adm/DCEWORK, is the pass one output file. The file contains commands that create new DCE principals.

-r  Runs pass two with a corrected DCE Error File, /opt/dcelocal/var/security/adm/DCEERS.

Usage

1. This utility enables interoperability between Resource Access Control Facility (RACF) and z/OS DCE and enables single sign-on for DCE users. It is used for cross linking information between RACF user IDs and the corresponding DCE principal IDs. Use mvsimpt when starting with the DCE Registry.

2. Before running the mvsimpt command, you must customize the file /opt/dcelocal/etc/IMPTVAR. See the z/OS DCE Administration Guide for a description of the files used by the mvsimpt command.

3. mvsimpt must be run from the z/OS UNIX System Services shell.

Examples

1. For pass one:

   mvsimpt -p1 -pw cellpw

   where cellpw is the cell administrator's password. This example creates a command file for pass two, using the RACF Unload file as input.

2. For pass one:
mvsimpt

mvsimpt -p1 -s /opt/dcelocal/var/security/adm/SEGMUNLD -pw cellpw
creates the command file for pass two, using a file name provided by the administrator instead of the
RACF Unload File.

3. For pass two:
   mvsimpt -p2 -r
   takes the DCE Error File, /opt/dcelocal/var/security/adm/DCEERS, (that has been corrected) and
rers runs pass two to add users to the DCE registry.

Related Information:  Commands:

   mvsexpt
   storepw

Books:

   z/OS DCE Administration Guide
pwd_strengthd

The sample Password Management server.

Format

```
pwd_strengthd [+/all[spaces] [+/alp]ha_num]] [-c]ache_size]] size [-m]in_len]] pwd_min_len
[-t]imeout]] minutes
```

Options

+all_spaces  Allow passwords to be all spaces. If this option is not set, the effective registry policy is used.

-all_spaces  Disallow passwords to be all spaces. If this option is not set, the effective registry policy is used.

+alpha_num  Allow passwords to consist only of alphanumeric characters. If this option is not set, the effective registry policy is used.

-alpha_num  Disallow passwords to consist only of alphanumeric characters. If this option is not set, the effective registry policy is used.

-cache_size size  Specify the number of hash buckets in the password cache. The password cache stores generated passwords that are retrieved when the password is strength checked. The password cache is a hash table with a linked list for collisions. Set the size to a reasonable value based on how large the cache will be on average. The default value is 100 if not specified.

-min_len pwd_min_len  Specify the minimum length of a password. If this option is not set, the effective registry policy is used.

-timeout minutes  Specify the time in minutes that generated passwords remain in the cache before they are deleted from memory. If this option is not specified, the default time is 30 minutes.

Usage

pwd_strengthd is a sample Password Management Server. It exports the rsec_pwd_mgmt application programming interface.

pwd_strengthd generates passwords and strength-checks them. It enforces the security registry policy for password strength-checking. Administrators can override the security registry policy using the command-line options (alpha_num, all_spaces, min_len.).

Administrators can subject principals to password-strength and -generation policies by attaching the following ERAs:

pwd_val_type  Specifies the password management policy the user must conform to when selecting passwords.

pwd_mgmt_binding  Specifies information required in order to connect to the password management server.

See the z/OS DCE Administration Guide for more information and examples.
rgyedit

Edits the registry database.

Format

\texttt{rgyedit}\ [[[-a | -p | -g | -o] [-s name \ [-up\{date\}]] [-v \ [name | -unix\_number number]] [-nq]] [-e] [-po] [-nr]]

\begin{center}
\textbf{Note to Users}
\end{center}

In the shell environment, the \texttt{rgyedit} command is entered as \texttt{rgy-edit}.

Options

You can specify only one of \texttt{-a}, \texttt{-p}, \texttt{-g}, or \texttt{-o}.

\begin{itemize}
\item \texttt{-a} \hspace{1cm} Edits or views accounts (default).
\item \texttt{-p} \hspace{1cm} Edits or views principals.
\item \texttt{-g} \hspace{1cm} Edits or views groups.
\item \texttt{-o} \hspace{1cm} Edits or views organizations.
\item \texttt{-s} \hspace{1cm} Uses the specified registry site.
\item \texttt{-up\{date\}} \hspace{1cm} Binds to a read-write registry site in the cell specified by the \texttt{-s} option.
\item \texttt{-v} \hspace{1cm} Views the registry entry specified by \texttt{name} or \texttt{-unix\_number number}. If no entry is specified, all entries are viewed.
\end{itemize}

Unless you specify the \texttt{-v} option, \texttt{rgyedit} operates interactively. In interactive mode, you enter the commands described in Subcommands For Principals, Groups, and Organizations on page 665.

\begin{itemize}
\item \texttt{-f} \hspace{1cm} Displays in full the entry (or entries) selected by the \texttt{-v} option. The full entry includes all fields except the membership list and organization policy.
\item \texttt{-unix\_number} \hspace{1cm} The number that corresponds to your name. Only valid with \texttt{-p} as UNIX ID, tied to the principal.
\item \texttt{-nq} \hspace{1cm} Specifies that delete operations will not be queried. The default is to prompt the user for verification when a delete operation is requested.
\item \texttt{-e} \hspace{1cm} Exit on error. The registry will end if an error is encountered.
\item \texttt{-po} \hspace{1cm} Display policy information for an organization view. (Views organizations only.) Only valid with \texttt{-o} and \texttt{-v}.
\item \texttt{-nr} \hspace{1cm} Disables automatic rebinding to a site. You must rebind to a site if you require new or different access to a currently bound site.
\end{itemize}

Usage

The \texttt{rgyedit} tool views and edits information in the registry database.

Changes made by \texttt{rgyedit} apply only to the registry. You can view and change only those registry objects to which you have been granted the appropriate permissions.
If you run **rgyedit** in batch mode, the input to the control programs and the output for each command are written to the job execution log. You can then determine which input commands were successful and which were not.

**Notes:**

1. The **rgyedit** command is supported in z/OS DCE. This function can also be performed by the **dcecp** commands **account**, **group**, **organization**, **principal**, **registry**, and **rgy_edit**. **dcecp** is the preferred interface.

2. If the password contains one or more blanks, do not specify the password on a subcommand. Instead, wait for the prompt.

**Starting rgyedit**

When you start **rgyedit** without the **-v** option, it displays the following prompt:

```
rgyedit=>
```

At this prompt, you can enter any of the **rgyedit** subcommands, and you are prompted for the required information. Or, you can enter the subcommand using the parameters specified with brackets ([ ] ) below. You may be prompted for any required information you do not enter.

**Subcommands For Principals, Groups, and Organizations**

In the **rgyedit** subcommands that follow, use two quotation marks with no spaces between them, to indicate a null **fullname**, **password**, **misc**, **homedir**, or **shell**, for example, "". Use quotation marks to embed spaces, quotation marks, or hyphens in **fullname**, **misc**, and **homedir**, if you specify the argument on the command line.

```
v[iew] [name | -u unix_number] [-f] [-m] [-po]
```

Views registry entries.

Whether **name** applies to a principal, group, or organization depends on the domain in which you run **rgyedit**. Use the **do[main]** subcommand to change domains.

If you specify the **-u unix_number** option, **rgyedit** displays all matching entries, including any aliases. If you do not specify **name** or **-u unix_number**, **view** displays a list of all the names in the current domain.

The **-f** option displays entries in full (all fields except the membership list and organization policy).

If you are viewing groups or organizations, **-m** displays the membership list. For principals, **-m** lists all groups of which the principal is a member, including groups that cannot appear in a project list.

If you are viewing organizations, **-po** displays policy information. If you do not enter the **-po** option, **rgyedit** shows only the organization's name and the UNIX number. **-po** is only valid in the organization domain.

```
a[dd] [principal_name [unix_num] [-f fullname] [-q quota]]
a[dd] [group_name [unix_num] [-f fullname] [-nl]] [-al]
a[dd] [organization_name [unix_num] [-f fullname]]
```

Creates a new name entry.

If you do not specify a **principal_name**, **group_name**, or **organization_name**, the **add** subcommand prompts you for a name. If you are adding organizations,
subcommand prompts you for policy information as well. If you specify only a
principal_name, group_name, or organization_name and no other arguments, the
object’s full name defaults to “” (that is, blank), the object’s UNIX number is assigned
automatically, and the object’s creation quota defaults to unlimited.

Use the -al option to create an alias for an existing principal or group. No two
principals or groups can have the same UNIX number, but a principal or group and all
its aliases share the same UNIX number. The -al option creates an alias name for a
principal or group and assigns the alias name the same UNIX number as the principal
or group. You will be prompted for the UNIX number, full name, and object information.

The -q option specifies the principal’s object creation quota, that is, the total number of
registry objects that can be created by the principal. If you do not specify this option,
the object creation quota defaults to unlimited (-1).

For groups, the -nl option indicates that the group is not to be included on project lists;
omitting this option allows the group to appear on project lists.

Changes a principal, group, or organization. Specify the entry to change with principal_name, group_name, or organization_name. If
you do not specify a principal_name, group_name, or organization_name, the change
subcommand prompts you for a name. If you do not specify any fields, the
subcommand prompts you for each field in succession. To leave a field unchanged,
press <RETURN> at the prompt. If you are changing organization entries in the
interactive mode, the subcommand prompts you for policy information as well.

For principals and groups, the -al option changes a primary name into an alias, and the
-pr option changes an alias into a primary name. This change can be made only from
the command line, not in the interactive mode. The primary must be deleted or
changed to an alias before the alias can be changed to primary.

For group entries, the -nl option indicates that the group cannot appear in project lists.
The -l flag allows the group to appear in project lists.

For organization entries, you can change policy information only in the interactive mode.
Changes to a principal name are reflected in membership lists that contain the principal
name. For example, if the principal ludwig is a member of the group composers and
the principal name is changed to louis, the membership list for composers is
automatically changed to include louis but not ludwig.

For reserved names, you can change only fullname.

When you change your DCE password, if you are enrolled in z/OS DCE single sign-on
support, you must use the storepw command to change your DCE password in the
RACF database. Failing to do so causes single sign-on to fail. See the storepw
command for more information.

Edits the membership list for a group or organization.

If you do not specify a group or organization, the member subcommand prompts you
for names to add or remove.
To add names or aliases to a membership list, use the `-a` option followed by the names separated by commas. To delete names from a membership list, use the `-r` option followed by the names separated by commas. If you do not include either the `-a` or `-r` option on the command line, `rgyedit` prompts you for names to add or remove.

Removing names from the membership list causes the account to be removed. Removing names from the membership list for a group or organization deletes the login account for a removed member (and, of course, eliminating any permissions granted as a result of the membership the next time the member’s ticket-granting ticket is renewed).

```
del[ete] name
```

Deletes a registry entry.

If you delete a principal, `rgyedit` deletes the principal's account. If you delete a group or organization, `rgyedit` deletes any accounts associated with the group or organization. You cannot delete reserved principals.

```
adopt uuid-string principal_name [-u unix_number] [-f fullname] [-q object_creation_quota]
adopt uuid-string group_name [-u unix_number] [-f fullname] [-nl]
adopt uuid-string org_name [-u unix_number] [-f fullname]
```

Creates a principal, group, or organization for the specified UUID.

The principal, group, or organization is created to adopt a registry object that cannot be accessed because it is owned by a UUID that is not associated with a principal or group, and no other principal, group, or organization has access rights to this object. UUIDs are associated with all registry objects when they are created, but when they are deleted, the association between the object and the UUID is also deleted.

The `principal_name`, `group_name`, or `organization_name` that you specify must be unique in the registry. Except for the way in which it is created, the principal, group, or organization created by the `adopt` subcommand is the same as any other principal, group, or organization.

The `uuid` option specifies the UUID to be assigned to the principal, group, or organization. This UUID must be the one that owns the orphaned object. The `uuid` consists of hexadecimal numbers in RPC string format as follows:

```
nnnnnnn-nnnn-nnnn-nnnnnnnnnnn
```

For cell principals only, the `-u` option specifies the UNIX number to be associated with the cell name. If you do not enter this option, the next sequential UNIX number is supplied as the default. For all principals other than cells, the UNIX number is extracted from information embedded in the principal's UUID and cannot be specified here.

For principals, the `-q` option specifies the principal's object creation quota. If you do not enter the option, the object creation quota is set to `unlimited`.

For groups, the `-nl` option turns off the project list inclusion property so that groups are not included in project lists. If you do not enter this option, the group is included in project lists.

For principals, groups, and organizations, the `-f` option supplies the object's full name. If you do not enter the `-f` option, the full name defaults to blank.

An error occurs if you specify a name or UNIX number that is already defined within the same domain of the database.

**Note:** In the current implementation of DCE, UNIX numbers are embedded in UUID numbers. If you try to create a group or organization to adopt an orphaned object and
fail, it may be caused by an embedded UNIX number that is no longer valid, because is
outside the range of valid UNIX numbers set for the cell as a registry property. Reset
the range of valid UNIX numbers to include the UNIX number embedded in the UUID
and try again to adopt the object.

Account Subcommands

view [pname [gname [oname]]] [-f]

Displays login accounts.

Without the -f option, view displays only the user fields in each account entry: principal,
group, organization, encrypted password, miscellaneous information, home directory,
and login shell.

With -f, view displays the full entry, including the administrative fields as well as the
user fields. Administrative information includes who created the account, when it was
created, who last changed it, when it was last changed, when it expires, whether it is
valid, whether the account principal's password is valid, and when the account
principal's password was last changed.

add [pname [-g gname -o org -mp password [-pw password | -rp]
[-m misc_info] [-h homedir] [-s shell] [-gsd good_since_date]
[-x account_exp_none | none] [-pv | -pnv | -av | -anv]
[-mcr lifespan] [-mcl lifespan] [-ena[ble] option | -dis[able] option]]

Creates a login account.

The -g and -o options specify the principal's group and organization. If the principal
specified in account is not already a member of the specified group and organization,
rgyedit automatically attempts to add the principal to the membership lists. If you do
not have the appropriate permissions for the group and organization, the attempt will fail
and the account will not be created.

The -rp option generates a random password for the account. The primary use of this
option is to create passwords for accounts that will not be logged into (because the
random password can never be supplied). The -pw option supplies a password for the
account on the command line.

The -mp option permits you to supply your password so your identity can be validated.
It must be specified if you have used either the -rp flag or the -pw option. In these
cases, however, the -mp option will not prompt you for your password.

If you supply a password in plain text, the system performs the encryption. The format
of password must adhere to the policy of the associated organization or the policy of
the registry as a whole, whichever is more restrictive.

Although the -s option is not used in the DCE MVS/ESA environment, it is included for
consistency between rgyedits across different operating systems. The information
supplied with the -m option creates the GECOS field in the account in the /etc/passwd
file . If you run the passwd_export command, this entry contains the
concatenation of the principal's full name and the information specified with the -m
option.

The -h option is similar to the -s option in that it is not used in the DCE MVS/ESA
environment, but is included for consistency between rgyedits across different
operating systems. The -h option specifies the path name of the principal's home
directory. The default homedir is / . The -s option specifies the path name of the
principal's login shell. The default shell is the null string.
The **-pnv** (password not valid) option warns the user that the password must be changed. At the next login (for a newly created account, the first login), the user must change the password.

**Note:** With this option, only an error is sent to the user. The changing of the password is not actually enforced. That is, it will still allow the user to login although he has not changed his password.

The **-pv** option indicates the password is currently valid and need not be changed (the default).

The **-x** option sets an expiration date for the account in yy/mm/dd/hh:ss format. The default is *none*, meaning that the account will never expire.

The **-anv** (account not valid) option specifies that the account is not currently valid for login. The **-av** option indicates the account is currently valid (the default).

The **-enable** and **-disable** options set or clear the following options:

- The **c[lient]** option, if enabled, allows the principal to act as a client and log in, acquire tickets, and be authenticated. If you disable **client**, the principal cannot act as a client. The default is enabled.

- The **s[erver]** option, if enabled, allows the principal to act as a server and engage in authenticated communication. If you disable **server**, the principal cannot act as a server that engages in authenticated communication. The default is enabled.

- The **po[stdated]** option, if enabled, allows tickets with a start time some time in the future to be issued to the account's principal. The default is disabled.

- The **f[orwardable]** option, if enabled, allows a new ticket-granting ticket (with a network address that differs from the present ticket-granting ticket address) to be issued to the account's principal. The default is enabled.

- The **pr[oxiable]** option, if enabled, allows a new ticket with a different network address than the present ticket to be issued to the account's principal. The default is disabled.

- The **T[GT_authentication]** option, if enabled, allows service tickets sent to the account's principal to use the standard DCE ticket-granting ticket authentication mechanism. The default is enabled.

- The **r[enewable]** option, if enabled, allows tickets issued to the principal’s ticket-granting ticket to be renewed. If this flag is enabled, the Security Service allows the principal’s ticket-granting ticket to be renewed for as long as its lifetime is valid. The default is enabled.

- The **dup[_session_key]** option allows tickets issued to the account's principal to have duplicate keys. The default is disabled.

The **-gsd** (good since date) is the date and time the account was last known to be valid. When accounts are created, this date is set to the account creation time. If you change the *good since date*, any tickets issued before the changed date are not valid. Enter the date in yy/mm/dd.hh:mm format.

The **-mcr** (maximum certificate renewable) option is the number of hours before a session with the principal’s identity expires, and the principal must log in again to reauthenticate. The default is 4 weeks.

The **-mcl** (maximum certificate lifetime) option is the number of hours before the Authentication Service must renew a principal’s service certificates. This flag requires no action on the part of the principal. The default is 1 day.
cfg -p pname -g gname -o oname -ng new_gname

Changes an account. The -p, -g, and -o options identify the account to change. The -ng, and -no options change the account's group and organization, respectively.

If you do not specify all three -p, -g, and -o options, wildcard updates can occur. For example, if you specify only the -g option, the changes affect all accounts that are associated with the named group. Note that you cannot use a wildcard to change passwords. To change a password, you must enter the -p, -g, and -o options.

All other options have the same meaning as described in the add command for accounts. Note that the -rp flag can be used to change the random passwords of the reserved accounts created by sec_create_db when the registry database is created.

del [-p pname] [-g gname] [-o oname]

Deletes the specified account.

Enter the -p option to delete the specified principal’s account. Enter the -g or -o option to delete accounts associated with the specified group or organization. If you enter the -g or -o option, rgyedit individually prompts to delete each account associated with the group or organization.


Creates a cross-cell authentication account in the local and foreign cells. This account allows local principals to access objects in the foreign cell as authenticated users and foreign cells to access objects in the local principal as authenticated users. The administrator in the foreign cell must have also set up a standard account, whose ID and password the administrator must supply to you.

The cellname variable specifies the full path name of the foreign cell with which you will establish the cross-cell authentication account. This name is stripped of the path qualifier and has a prefix of krbtgt. The resulting name is used as the primary name for the cross-cell authentication account. For example, if you enter /.../dresden.com, the principal name is krbtgt/dresden.com.

The -ul option specifies the UNIX number for the local cell’s principal, and the -uf option specifies the UNIX number for the foreign cell’s principal. If you do not specify these UNIX numbers, they are generated automatically.

The -gl and -ol options specify the local account’s group and organization. The -gf and -of options specify the foreign account’s group and organization.

The -mp option specifies the password of the person who started rgyedit.

The -fa option specifies the name identifying the account in the foreign cell, and the -fp option specifies the account’s password.

The -q option specifies the total number of objects that can be created in your cell’s registry by all foreign users who use the cross-cell authentication account to access your cell. The object creation quota defaults to 0 (zero), meaning that principals in the foreign cell cannot create objects in the local cell. The object creation quota set for
your cell’s account in the foreign cell places the same restriction on the number of objects that your cell’s principals can create in the foreign cell’s directory.

The `-x` option specifies the account expiration date for both the local and foreign accounts. The default for this option is `none`.

Note that the object creation quota for the local account defaults to 0 (zero), meaning that principals in the foreign cell cannot create objects in the local cell. You can change this state with the `rgyedit change` subcommand.

---
**Important Note to Users**

Careful cleanup must be done when reconfiguring intercell communications. Use the following information when applicable.

If you reconfigure one cell or accidentally remove a cross-cell account, you must perform some cleanup before intercell communication is established. This will be evident when you try to access the foreign site using the following command:

```
rgy_edit⇒ si /.../dino
```

the system will respond with

```
?(rgy_edit) Unable to open the registry at site “/.../dino” - Registry server unavailable (Registry Edit Kernel) (dce / sad)
Current site is: registry server at <none>
```

To perform the cleanup, do the following:

1. In both cells, remove the surrogate authentication principal under `krbtgt/<foreign_cell>`. Use the following example:

   ```
   # dce_login cell_admin -dce-
   # rgy_edit
   rgy_edit>do p
   Domain changed to: principal
   rgy_edit>v
   .
   .
   cell_admin 1RzerodotRzerodot
   krbtgt/dino 1Rzerodot1
   hosts/sys2/self 1Rzerodot2
   hosts/sys2/cds-server 1Rzerodot3
   hosts/sys2/dfs-server 1Rzerodot5
   hosts/sys2/gda 1Rzerodot8
   krbtgt/sauro 1Rzerodot6
   rgy_edit>del krbtgt/sauro <-------------- delete this surrogate principal
   rgy_edit>quit
   ```

2. In one cell, enter the `rgy_edit cell` subcommand again to generate the accounts and principals. It then recreates the peer trust between cells. Both registers are modified to include the new cross-cell accounts.

3. Ensure that the credentials of all principals do not have expired tickets. You can verify this using the `klist -e` command and noting the tickets for both local and remote cells. If the principal has expired tickets, it cannot communicate with the foreign cell. Use the following example:
#klist -e

DCE Identity Information:

Global Principal: /.../dino/hosts/sys2/self
Cell: 00795a36-3ff8-1d0f-b041-10005aa8c755 /.../dino
Principal: 00000066-3ff9-2d0f-b000-10005aa8c755 hosts/sys2/elf
Group: 0000000c-3ff9-2d0f-b001-10005aa8c755 none
Local Groups:
   0000000c-3ff9-2d0f-b001-10005aa8c755 none
   0000006b-400f-2d0f-9101-10005aa8c755 sybsys/dce/dts-servers

Identity Info Expires: 96/10/05:19:24:38
Account Expires: never
Passwd Expires: never

Kerberos Ticket Information:
Ticket cache: /opt/dcelocal/var/security/creds/dcecred_ffffffff
Default principal: hosts/sys2/self@dino
Server: krbtgt/dino@dino
   valid 96/10/66:06:28:01 to 96/10/06:16:28:01
Server: dce-rgy@dino
   valid 96/10/66:06:28:01 to 96/10/06:16:28:01
Server: dce-ptgt@dino
   EXPIRED; was valid 96/10/05:06:38:24 to 96/10/05:08:38:24
Client: dce-ptgt@dino Server: krbtgt/dino@dino
   EXPIRED; was valid 96/10/05:06:38:24 to 96/10/05:08:38:24
Client: dce-ptgt@dino Server: hosts/sys2/cds-server@dino
   EXPIRED; was valid 96/10/05:06:38:24 to 96/10/05:08:38:24
Client: dce-ptgt@dino Server: dce-rgy@dino
   EXPIRED; was valid 96/10/05:06:38:24 to 96/10/05:08:38:24
Client: dce-ptgt@dino Server: dce-rgy@dino
   EXPIRED; was valid 96/10/05:06:59:35 to 96/10/05:08:38:24
Server: dce-ptgt@dino
   valid 96/10/06:08:39:26 to 96/10/06:10:39:26
Client: dce-ptgt@dino Server: krbtgt/dino@dino
   valid 96/10/06:08:39:26 to 96/10/06:10:39:26
Client: dce-ptgt@dino Server: hosts/sys2/cds-server@dino
   valid 96/10/06:08:39:26 to 96/10/06:10:39:26
Client: dce-ptgt@dino Server: dce-rgy@dino
   valid 96/10/06:09:00:38 to 96/10/06:10:39:26

In this example there are some EXPIRED tickets. To remove the expired tickets that belong to an interactive user, run the kinit command. You can replace the credentials entirely by using the kdestroy command and then performing another dce_login. If the credentials belong to a non-interactive principal (like those listed above, which belong to the machine principal self) you must destroy the credentials and recreate them. For the machine principal, stop the dced process and start it again.

Until you remove old tickets, the principal using those credentials will not be able to communicate with the foreign cell, and operations such as rgy_edit site /.../<foreign_cell> and cdscp show cell /.../<foreign_cell> will fail.

Key Management Subcommands

The key management subcommands must be in command-line mode.

kta[dd] -p principal [-a[uto] | -pw password] [-r[egistry]] [-f keyfile]
rgyedit

Creates a password for a server or machine in the keytab file on the local node.

The `-p` option specifies the name of the server or machine principal for which you are creating a password.

The `-pw` option lets you supply the password on the command line. If you enter this option or the `auto` option, the `ktadd` prompts for the password.

The `-a` option generates the password randomly. If you use this option, you must also use the `-r` option.

If you do not specify the `-auto` or the `-pw` option, you are prompted for a password.

If you want to use an existing password, you can specify it with the `-pw` option. If you do not specify either of the `-a` or `-pw` options, you are prompted to enter a password.

The `-r` option updates the principal's password in the registry to match the string you enter (or automatically generate) for the password in the keytab file. Use it to ensure that when you change a principal's password in the keytab file, the principal's password in the registry is kept identical to the principal's password in the keytab file. To use this option, a password for the principal must exist in the default keytab file or the keytab file named by the `-f` option.

The `-f` option specifies the name of the server key file on the local node to which you are adding a password. If you do not specify a keytab file name, `/krb5/v5srvtab` is used. You must have proper access rights to update entries in the keytab files. In DCE MVS/ESA, the specified file will be converted to an MVS data set name as follows:

- Each directory in the UNIX file path name will be truncated to 8 alphanumeric characters.
- Only the last 4 elements in the path name are kept.
- The user's z/OS ID is added as a prefix to these four elements to create the eventual 5 qualifier (maximum) z/OS data-set name.

No attempt is made to ensure the z/OS data-set name is unique. It is the user's responsibility to ensure the first 8 characters in the last 4 elements of the UNIX file name are unique so no collisions occur. The following is an example of this concept.

z/OS userid: TSDSUSR
UNIX filenme: /home/development/temp/server/keytab.files/tmp_appl.1
z/OS data set name: TSDSUSR.TEMP.SERVER.KEYTABFI.TMPAPPL1

The user is also responsible for ensuring that each element begins with an alphabetic character so that the eventual z/OS data set name is a valid one.

```bash
ktlist [-p principal_name] [-f keyfile]
```

Displays principal names and password version numbers in the local keytab file. The `-p` option specifies the name of the server or machine principal for which you are displaying password version numbers.

For information regarding the `-f` option, refer to the `ktadd` command.

```bash
ktdelete -p principal -v version-number [-f keyfile]
```

Deletes a server or machine principal's password entry from a keytab file on the local node.

The `-p` option is the name of the server or machine principal for whom you are deleting a password entry.

The `-v` option specifies the version number of the password you want to delete.

Version numbers are assigned to a principal's password whenever the password is
changed. Any servers or machines still using tickets granted under the old password can run without interruption until the ticket expires at the end of its lifespan.

For information regarding the `-f` option, refer to the `ktadd` command.

Miscellaneous Commands

domain [ p[principal] | g[roup] | o[rganization] | a[ccount]]
  Changes or displays the type of registry information being viewed or edited.
  You can specify `p` for principals, `g` for groups, `o` for organizations, or `a` for accounts. If you supply no argument, `rgyedit` displays the current domain.

site [[name] | -[u]pdate | -[q]ery]
  Changes or displays the registry site being viewed or edited.
  The `name` variable is the fully qualified name of the cell that contains the registry to which you want access. If you supply no argument, `rgyedit` displays the current site.
  The `-u` option indicates you want to talk to an update site in the specified cell. The `-q` option indicates you want a query site.

properties
  Changes or displays registry properties.
  This command prompts you for changes. Press `<RETURN>` to leave information unchanged.

policy [organization_name] [-al lifespan | forever]
  [-pl passwd_lifespan | forever] [-px passwd_exp_date | none]
  [-pm passwd_min_length] [-pa | -pna] [-ps | -pns]
  Changes or displays registry standard policy or the policy for an organization.
  Enter `organization_name` to display or change policy for that specific organization. If you do not enter `organization_name`, the subcommand affects standard policy for the entire registry.
  The `-al` option determines the account’s lifespan, the period during which accounts are valid. After this period of time passes, the accounts are not valid and must be re-created. An account’s lifespan is also controlled by the `-x` option of the `add` and `change` subcommands. If the two lifespans conflict, the shorter one is used. Enter `lifespan` in the following format:
  
  weekswdaysdhourshminutes

  For example, 4 weeks and 5 days are entered as `4w5d`.

  If you enter only a number and do not designate weeks, days, or hours, the designation defaults to hours. If you end the lifespan with a number without a weeks, days, or hours designation, the number with no designation defaults to seconds. For example, `12w30` is assumed to be 12 weeks, 30 seconds.

  The `-pl` option determines the password lifespan, the period of time during which account passwords are valid. After this period, the passwords expire and must be changed. Enter `passwd_lifespan` as a number indicating the number of days the password is valid. If you define a password lifespan as `forever`, the password has an unlimited lifespan.

  The `-px` option specifies password expiration dates in `yy/mm/dd/hh.mm.ss` format. If you define a password expiration date as `none`, the password has an unlimited lifespan.

  The `-pm`, `-ps`, `-pns`, `-pa`, and `-pna` options all control the format of passwords as follows:
- **-pm** specifies the minimum length of passwords in characters. If you enter 0, no password minimum length is in effect.

- **-ps** and **-pns** specify whether passwords can contain all spaces (-ps) or cannot be all spaces (-pns).

- **-pa** and **-pna** specify whether passwords can consist of all alphanumeric characters (-pn) or must include some non-alphanumeric characters (-pna).

**au[thpolicy]** Changes or displays registry authentication policies.

This command prompts you for changes. Press <RETURN> to leave information unchanged.

**def[aults]** Changes or displays the home directory, login shell, password valid option, account expiration date, and account valid-option default values that **rgyedit** uses.

This command first displays the current defaults. It then prompts you for whether you want to make changes. If you make changes, **defaults** immediately changes the defaults for the current session, and saves the new defaults in **/rgy_editrc**. The newly saved defaults are used until you change them.

**h[elp] [command]**

Displays usage information for **rgyedit**.

If you do not specify a particular command, **rgyedit** lists the available commands.

**q[uit]** Exit from **rgyedit**.

**e[xit]** Exit from **rgyedit**.

**l[ogin]** Lets you establish a new network identity for use during an **rgyedit** session. The **rgyedit** login command prompts for a principal name and password.

**sc[ope] [name]** Limits the scope of the information displayed by the **view** subcommand to the directory (specified by **name**) in the registry database.

**Related Information**

Commands:

**storepw**
secadmin

The registry replica administration tool.

Format

```bash
secadmin [-site name] [-nq]
```

--- Note to Users

In the shell environment, enter the `secadmin` command as `sec_admin`.

**Options**

- **-site name**  
The `-site` option causes `secadmin` to bind to the replica specified by the `name` argument. If the option is not used, `secadmin` binds randomly to any replica in the local cell.

  The `name` argument can be:
  
  - A specific cell name (or `/:` for the local cell) to bind to any replica in the named cell.
  - The global name of a replica to bind to that specific replica in that specific cell.
  - The name of the replica as it appears on the replica list to bind to that replica in the local cell.
  - A string binding to a specific replica; for example, a string binding is `ncadq_ip_udp:15.22.144.163`. This form is used primarily for debugging or if the Cell Directory Service is not available.

- **-nq**  
The `-nq` flag turns off queries initiated by certain `secadmin` subcommands before they perform a specified operation. For example, the `delrep` subcommand deletes a registry replica. Before `secadmin` performs the deletion, it prompts for verification. If you start `secadmin` with the `-nq` option, the subcommand performs the deletion without prompting.

**Usage**

The registry database is replicated: each instance of a registry server, `secd`, maintains a working copy of the database in virtual memory and on disk. One server, called the master replica, accepts updates and handles the subsequent propagation of changes to all other replicas. All other replicas are slave replicas, which accept only queries. Each cell has one master replica and may have numerous slave replicas.

Using the `secadmin` command you can:

- View a list of replicas
- Delete a replica
- Reinitialize a replica
- Stop a replica
- Put the master replica into and out of the maintenance state
- Generate a new master key used to encrypt principal keys
- Turn the master registry into a slave registry and a slave registry into the master registry
- Control the audit function on a z/OS UNIX System Services Security server

Note that `secadmin` does not add, delete, or modify information in the database, such as names and accounts. Use the `dcecp registry` command to modify registry database entries.
Note: With the exception of the following commands, the secadmin command is supported in z/OS DCE by the dcecp registry command.

- audit
- exit
- help
- monitor
- quit

The Default Replica and Default Cell

Most secadmin commands are directed to a default replica. When secadmin is started, it automatically binds to a replica in the local cell. This replica becomes the default replica.

Identifying the Default Replica and the Default Cell: You use the site subcommand to change the default replica and, optionally, the default cell. When you use the site command, you can supply the name of a specific replica, or you can simply supply the name of a cell. If you supply a cell name, secadmin binds to a replica in that cell randomly. If you supply a specific replica name, secadmin binds to that replica.

Specifically, you can supply any of the following names to the site subcommand:

- Cell name: If you enter a cell name, the named cell becomes the default cell. The secadmin command randomly chooses a replica to bind to in the named cell, and that replica becomes the default replica.
- Global name: Given to the replica when it was created, a global name identifies a specific replica in a specific cell. That cell becomes the default cell and that replica the default replica.
- Replica name: As it appears on the replica list (a list maintained by each Security Server containing the network addresses of each replica in the local cell). That replica becomes the default replica and the cell in which the replica exists becomes the default cell.
- Network address: Address of the host on which the replica is running. The replica on that host becomes the default replica, and the cell in which the host exists becomes the default cell.

Naming the Default Replica: As an example, assume a replica named subsys/dce/sec/rs_server_250_2:

- Exists in the local cell /./dresden.com
- Has a global name of /.../dresden.com/subsys/dce/sec/rs_server_250_2
- Is named subsys/dce/sec/rs_server_250_2 on the replica list
- Runs on a host whose ip network address is 15.22.144.248

This replica can then be identified to the site subcommand in any of the following ways:

- /.../dresden.com/subsys/dce/sec/rs_server_250_2 — The replica's full global name.
- subsys/dce/sec/rs_server_250_2 — The replica's cell-relative name on the replica list.
- ncadg_ip_udp:15.22.144.248 — The network address of the host on which the replica runs.
**Naming the Default Cell:** When a default replica is identified specifically, its cell becomes the default cell. In the example above, the default cell is `/.../dresden.com`.

You can specify simply a cell name to the `site` subcommand. When this is done, any replica in that cell is selected as the default replica.

For example, assume

`/.../bayreuth.com/subsys/dce/sec/rs_server_300_1`

and

`/.../bayreuth.com/subsys/dce/sec/rs_server_300_2`

are replicas in the cell `/.../bayreuth.com`.

If you type

```bash
site /.../bayreuth.com
```

then

`/.../bayreuth.com`

becomes the default cell and either

`/.../bayreuth.com/subsys/dce/sec/rs_server_300_1`

or

`/.../bayreuth.com/subsys/dce/sec/rs_server_300_2`

becomes the default replica.

**Automatic Binding to the Master**

Some of the `secadmin` subcommands can act only on the master registry and require binding to the master registry. If you run a subcommand that acts only on the master and the master is not the default replica, `secadmin` attempts to bind to the master replica in the current default cell automatically. If this attempt is successful, `secadmin` displays a message that the default replica was changed to the master registry. The master registry then remains the default replica until you change it with the `site` subcommand. If the attempt to bind is not successful, `secadmin` displays an error message, and the subcommand fails.

**Starting secadmin**

When you start `secadmin`, it displays the current default replica's full global name and the cell in which the replica exists. Then it displays the `sec_admin>` prompt.

```bash
$ secadmin
Default replica: /.../dresden.com/subsys/dce/sec/music
Default cell: /.../dresden.com
sec_admin>
```

Enter any of the `secadmin` subcommands at the `sec_admin>` prompt.
Subcommands

The subcommand descriptions that follow use default_replica to indicate the default replica and other_replica to indicate a replica other than the default. other_replica must identify a replica in the default cell. It is specified by its name on the cell's replica list (that is, by its cell-relative name). Use the lrep subcommand to view the default cell's replica list.

```plaintext
```

The -enable option starts the auditing function in the default replica.

The -disable option stops the auditing function in the default replica.

The -show option displays the state of the audit function in the default replica.

The -rewind option moves the pointer where the next audit record will be written to the beginning of the audit file. Any records previously written to this file are no longer accessible.

The -switch option closes the currently active audit file and starts a new one.

This subcommand only works on z/OS UNIX System Services security servers. Security servers on other platforms return a status code specifying that the interface is not known.

```plaintext
become [-master] | [-slave]
```

The -master option makes the current default replica (which must be a slave) the master replica.

The -slave option makes the current default replica (which must be the master) a slave replica.

This method of changing to master or slave can cause updates to be lost. The change_master subcommand is the preferred means of designating a different master replica. However, you may find the become -master command useful if the master server is irrevocably damaged and you are unable to use change_master.

```plaintext
change_master -to other_replica
```

Makes the replica specified by other_replica, the master replica. To perform this operation, other_replica must be a slave, and the current default replica must be the master. If the current default replica is not the master, secadmin attempts to bind to the master.

If the change operation is successful, the current master:

1. Applies all updates to other_replica
2. Becomes a slave
3. Tells other_replica to become the master

```plaintext
delrep other_replica [-force]
```

Deletes the registry replica identified by other_replica. To perform this operation, the current default replica must be the master. If it is not, secadmin attempts to bind to the master.

If the delete operation is successful, the master:

1. Marks other_replica as deleted
2. Propagates the deletion to all replicas on its replica list
3. Delivers the delete request to other_replica
4. Removes other_replica from its replica list

The -force option causes a more drastic deletion. It causes the master to first delete other_replica from its replica list and then to propagate the deletion to the replicas that remain on its list. Because this operation never communicates with the deleted replica,
use the **-force** option only when the replica has died and cannot be recovered. If you use
*force* while *other_replica* is still running, then use the *destroy* subcommand to eliminate
the deleted replica.

**help [command]**

Lists the **secadmin** subcommands and shows their allowed abbreviations. If *command* is
specified, displays help for the specified command.

**info [-full]**

Displays status information about the default replica.

The **info** subcommand contacts the default replica to obtain the appropriate information. If
this information is not available, **info** displays the replica name with a message stating the
information is not available.

Without the **-full** option, **info** displays:

- The default replica's name and the name of the cell in which the replica exists
- Whether the replica is a master or a slave
- The replica's state
- The date and time the replica was last updated and the update sequence number
- An indication of the replica's state, as follows:
  - **Bad State** — The state of the replica prohibits the requested operation.
  - **Uninitialized** — The database is a stub database that was not initialized by the
    master replica or another up-to-date replica.
  - **Initializing** — The replica is in the processes of being initialized by the master replica
    or another up-to-date replica.
  - **In Service** — The replica is available for use
  - **Saving Database** — The replica is in the process of saving its database to disk
  - **Copying Database** — The replica is in the process of initializing (copying its database
to) another replica
  - **In Maintenance** — The replica is unavailable for updates but will accept queries
  - **Changing Master Key** — The replica is in the process of having its master key
    changed
  - **Becoming Master** — The replica is in the process of becoming the master replica
    (applicable to slave replicas only).
  - **Becoming Slave** — The master replica is in the process of becoming a slave replica
    (applicable to master replicas only).
  - **Closed** — The replica is in the process of stopping.
  - **Duplicate Master** — The replica is a duplicate master and will be deleted.
  - **Deleted** — The replica is in the process of deleting itself.

The master replica is available for

- Queries, when it is in the in-service, copying-database, in-maintenance, master-key-changing, and
  becoming-slave states.
- Updates, only when it is in the in-service state.

A slave replica is

- Available for queries, when it is in the in-service, copying-database, master-key-changing, and
  becoming-slave states.
- It accepts updates from the master replica only when it is in the in-service state.
- It accepts a request from the master replica to initialize only when it is in the uninitialized or in-service state.

The -full option displays all the above information and the following information:

- The default replica’s unique identifier
- The replica’s network addresses
- The unique identifier of the cell’s master replica
- The network addresses of the cell’s master replica
- The master sequence number, which is the sequence number of the event that made the replica the master
- The update sequence numbers that are still in the propagation queue and have yet to be propagated (if the replica is the master replica)
- The DCE software version number

init[ep] other_replica

Reinitializes a replica by copying an up-to-date database to other_replica.

The master replica initiates and guides the operation. If the operation is successful:

1. The master replica:
   a. Marks other_replica for reinitialization
   b. Tells other_replica to reinitialize itself
   c. Gives other_replica a list of replicas with up-to-date databases

2. The other_replica picks a replica from the list and asks that replica to initialize it (that is, to copy its database to other_replica)

To perform this operation, other_replica must be a slave, and the current default replica must be the master. If the current default replica is not the master, secadmin attempts to bind to the master.

Generally, this subcommand is not used under usual conditions.

lr[ep] [-s[tate]] [-u[uid]] [-a[dr] [-p[rop] [-a[l]l]]]

Lists the replicas on the default replica’s replica list.

If you enter no options, the display includes the replica name and whether it is the master replica. In addition if the master replica’s list is being displayed, slave replicas marked for deletion are noted. With options, the display includes this information and the information described in the following paragraphs.

The -state option shows each replica’s current state, the date and time the replica was last updated, and the update sequence number.

To obtain this information, lrep contacts each replica. If this information is not available from the replica, lrep prints the replica name and a message stating the information is not available.

The -addr option shows each replica’s network addresses.

The -uuid option shows each replica’s unique identifier.

The -prop option shows:
- The date and time of the last update the master sent to each slave replica
- The sequence number of the last update
secadmin

- The number of updates not yet applied to each slave replica
- The status of the master replica’s last communication with each slave replica
- The propagation state of each slave replica. This state illustrates how the master replica views the slave replica and can be any of the following:
  - Bad State — The state of the replica prohibits the requested operation
  - Marked for Initialization — The replica was marked for initialization by the master replica
  - Initialized — The replica was initialized by the master replica
  - Initializing — The replica is in the process of being initialized by the master replica
  - Ready for Updates — The replica was initialized by the master replica and is now available for propagation updates from the master replica
  - Marked for Deletion — The replica was marked for deletion by the master replica

This information is obtained from the master replica; the slave replicas are not contacted for this information.

The -prop option is valid only for the master.

For slave replicas, the -all option shows all the information above except that displayed by the -prop option. For the master replica, the -all option shows all the information.

**mas[ter_key]** Generates a new master key for the default replica and re-encrypts account keys using the new key. The new master key is randomly generated.

Each replica (master and slaves) maintains its own master key used to access the data in its copy of the database.

**monitor [-r m]** Periodically lists the registry replicas stored in the current default replica’s replica list. The list includes each replica’s current state, the date and time the replica was last updated and the update sequence number.

The monitor subcommand contacts each replica to obtain the information it displays. If this information is not available from the replica, monitor displays the replica name and a message stating that the information is not available.

The -r option causes the replicas to be listed at intervals you specify. m is a number of minutes between intervals. The default is 15 minutes.

**destroy default_replica**

Destroys the current default replica. To perform this operation, the current default replica and the default replica you name as default_replica must be the same. This is to confirm you want to perform the deletion.

If the operation is successful, the default replica deletes its copy of the registry database and stops running. This subcommand does not delete default_replica from the replica lists. Use the delrep -force subcommand to delete the replica from the other replica lists.

The preferred way to delete replicas is to use the delrep subcommand. However, the destroy subcommand can be used if delrep is unusable because the master is unreachable or the replica is not on the master’s replica list.

**site [name [-u[update]]]**

Sets or displays the default cell and the default replica.

The name argument identifies the replica to set as the default replica and, as a result, the default cell. It can be:
• A specific cell_name (or /.: for the local cell) to make any replica in the named cell the default.

• The global name of a replica to make the specified replica in the specified cell the default.

• The name of a replica as it appears on the replica list to make the named replica (which exists in the default cell) the default replica.

• A string binding to a specific replica. An example of a string binding is `ncadg_ip_udp:15.22.144.163`. This form is used primarily for debugging or if the Cell Directory Service is not available.

The `-u` option specifies that `secadmin` should find the master replica. You usually specify the name of a cell for `name` in conjunction with the `-u` option. In this case, `secadmin` finds the master replica in that cell. If you use a replica name for `name`, `secadmin` queries the named replica to find the master replica in the named replica's cell.

If you supply no arguments, `secadmin` displays the current default replica and default cell.

`stop` Stops the Security Server (`secd`) associated with the default replica.

`stat[e] -maintenance | -service` Puts the master replica into maintenance state or takes it out of maintenance state. This subcommand is useful for performing backups of the registry database.

If the current default replica is not the master, `secadmin` attempts to bind to the master.

The `-maintenance` flag causes the master replica to save its database to disk and refuse any updates.

The `-service` flag causes the master replica to return to its normal “in service” state and start accepting updates.

`e[xit] or q[uit]` The `quit` and `exit` subcommands end the `secadmin` session.

**Examples**

1. The following example starts `secadmin` and uses the `lrep` subcommand to list replicas on the replica list and their states:
secadmin

$ /bin/sec_admin
  Default cell: /.../dresden.com
  Default replica: /.../dresden.com/subsys/dce/sec/rs_server_ .250_2

sec_admin> lrep -st

  Default cell: /.../dresden.com
  Default replica: /.../dresden.com/subsys/dce/sec/rs_server_ .250_2

  subsys/dce/sec/master (master)
    state: in service - master
    Last update received at: Tue Sep 5 14:17:43 1995
    Last update's seqno: 0.1974

  subsys/dce/sec/rs_server_250_2
    state: in service - slave
    Last update received at: Tue Sep 5 14:17:43 1995
    Last update's seqno: 0.1974

  subsys/dce/sec/rs_server_250_3
    state: in service - slave
    Last update received at: Tue Sep 5 14:17:43 1995
    Last update's seqno: 0.1974

sec_admin>

2. The following example sets the default replica to the master in the local cell:

    sec_admin> site .: -u
    Default cell: /.../dresden.com
    Default replica: /.../dresden.com/subsys/dce/sec/master

    sec_admin>
sec_create_db

Starts the registry database creation utility.

Format

```
sec_create_db [-master | -slave] -my[name] my_server_name
[-rdb | -hfs]
[-k[eyseed] keyseed] [-c[reator] creator_name]
[-p[assword] creator_password]
[-c[u]nix_id creator_unix_number] [-u[uid] cell_uuid]
[-p[e]rson_low_unix_id person_low_unix_id]
[-g[roup_low_unix_id] group_low_unix_id]
[-o[rg_low_unix_id] org_low_unix_id]
[-m[a]x_unix_id] max_unix_id] [-force]
```

Options

- **-master | -slave** Specifies whether the `sec_create_db` command should create a registry database for a master replica (`-master`), or for a slave replica (`-slave`). All of the `sec_create_db` options can be used with the `-master` option. Only the `-myname`, `-keyseed`, and `-verbose` options can be used with the `-slave` option. A master registry database is created if neither `-master` nor `-slave` is specified.

- **-myname** Specifies the replica name used by the DCE Directory Service to locate the machine on which this security server will be running. For DCE/MVS, the replica name should be in the form:

  ```
  subsys/dce/sec/name
  ```

  where the `name` component of the directory path can be any value that is unique within the cell.

- **-rdb | -hfs** Specifies whether the registry database is created using the DB2® relational database (`-rdb`) or the hierarchical file system (`-hfs`). This hierarchical file system is used if this option is not specified.

- **-keyseed** Specifies a character string to initialize the random key generator used to create the registry database master key. The master key is used to encrypt account passwords. Each replica has its own master key.

  You are prompted to enter the keyseed if this option is not specified on the command line.

- **-creator** Specifies the principal name of the initial privileged user of the registry database (known as the registry creator).

  The registry creator is set to `root` if this option is not specified.

- **-password** Specifies the default password for the `root` and registry creator accounts. The default password is set to `-dce-` if this option is not specified.

- **-cunix_id** Specifies the UNIX number assigned to the registry creator principal.

  The UNIX number is dynamically assigned if this option is not specified.

- **-uuid** Specifies the UUID for the cell. The cell UUID is dynamically generated if this option is not specified.
sec_create_db

-person_low_unix_id
   Specifies the starting UNIX number for principals.
   The initial UNIX number is set to 100 if this option is not specified.

-group_low_unix_id
   Specifies the starting UNIX number for groups.
   The initial UNIX number is set to 100 if this option is not specified.

-org_low_unix_id
   Specifies the starting UNIX number for organizations.
   The initial UNIX number is set to 100 if this option is not specified.

-max_unix_id
   Specifies the maximum UNIX number that can be assigned.
   The maximum UNIX number is set to 32767 if this option is not specified.

-force
   The sec_create_db command returns an error if a registry already exists. The -force option causes the sec_create_db command to overwrite an existing registry, causing the loss of all existing registry definitions.

Usage

The sec_create_db creates a registry database in the /dcelocal/var/security/rgy_data directory on the local system. An error is reported if a registry database already exists in this directory. Normally, the registry database is created only once by the system configuration tool, dceconf. You can use the sec_create_db command if you have to re-create the registry database. You must be root to run the sec_create_db command.

The registry database created for a master replica is initialized with the principals, groups, organizations, and accounts that are necessary to start the security server. When the security server is started, you need to use the dcecp command.

The registry database created for a slave replica is a minimal database. When the security server is started, it will contact the master security server for the cell in order to initialize the replica database from the master database.

In addition to the registry database, the sec_create_db command creates (-master) or updates (-slave) the dcelocal/etc/security/pe_site file. This file contains the binding information needed by the dced daemon to contact the security server during DCE kernel initialization. This file is also used by the security client to bind to the security server when the Directory Name Service is not available.

Before re-creating the registry database for a slave replica, use the dcecp registry delete command with the -force option to ensure that the replica was removed from the replica list maintained by the master security server.

If you re-create the registry database for the master security server, the new registry must be compatible with the existing cell configuration. One way to ensure this is to use the sec_export_db command to export the registry, use the sec_create_db command to re-create the registry, then use the sec_import_db command with the -replace option to import the previous registry definitions.
Examples

This example creates a new master security replica with the following properties:

- Replica name of `subsys/dce/sec/master`
- Creator name of `cell_admin` with the password `dce2dce`
- Master key generated using the seed `dcemvs`
- The maximum UNIX ID that can be assigned will be 65535

```
sec_create_db -master -myname subsys/dce/sec/master
              -creator cell_admin -password dce2dce -keyseed dcemvs
              -max_unix_id 65535
```
Starts the Security server daemon.

**Format**

```
secd [-locksm[ith] pname [-lockpw] [-rem[ote]] [-b[ootstrap]] [-restore_master] [-v[erbose]]
```

**Options**

The following options all start the security server on the local node:

- **-locksm[ith]** Replaces the Security Server in locksmith mode. Use this mode if you cannot access the registry as the principal with full registry access.

  - **pname** The `pname` argument is the name of the locksmith principal. If no registry account exists for this principal, `secd` will prompt you for one.

- **-lockpw** Prompt for a new locksmith password when running in locksmith mode. You can specify a new password for the locksmith account when the old one is unknown.

- **-rem[ote]** Allows the locksmith principal to log in remotely. If this option is not used, the principal must log in from the local machine on which `secd` will be started.

- **-b[ootstrap]** Always waits 1 minute between tries to export binding information to the Cell Directory Service during DCE configuration. If you do not specify this option, `secd` sleeps for 1 minute during initialization if DCE is not available when it tries to export binding information. If the export fails a second time, it sleeps for 2 minutes before it tries again. If it still fails, it sleeps for 4, 8, and 16 minutes between retries. Then, sleep time stays at 16 minutes until the binding export succeeds.

- **-restore_master** Marks all slave replicas for initialization during the master restart. Use this option only to recover from a catastrophic failure of the master security server; for example, when the database is corrupted and then restored from a backup tape.

- **-v[erbose]** Runs in verbose mode.

**Usage**

The `secd` daemon is the security server. It manages all access to the registry database. You must have operator authority to start `secd`.

For information on how to start and stop the `secd` process, see the Starting and Stopping z/OS DCE section in the [z/OS DCE Administration Guide](link).

**Locksmith Mode:** The `secd locksmith` option starts `secd` in locksmith mode. In locksmith mode, the principal name you specify for `secd` with `pname` becomes the locksmith principal. As the locksmith principal, you can repair deliberate or accidental changes that prevent you from logging in with full registry access privileges.

If no account exists for `pname`, `secd` establishes one and prompts you for the account's password. (Use this password when you log in to the account as the locksmith principal). If an account for `pname` exists, `secd` changes the account and policy information as described in Table 17 on page 689 and Table 18 on page 689. These changes ensure that even if account or registry policy was tampered with, you will now be able to log in to the locksmith account.
In locksmith mode, all principals with valid accounts can log in and operate on the registry with normal access checking. The locksmith principal, however, is granted special access to the registry: no access checking is performed for the authenticated locksmith principal. As the locksmith principal, you can operate on the registry with full access.

Notes:

1. The locksmith principal or account should never be modified during a locksmith session.
2. Always end the `secd` command by issuing a `kill -15` command to the process that is running `secd`.

<table>
<thead>
<tr>
<th>Table 17. Locksmith Account Changes Made by the Security Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If the Security Server finds</strong></td>
</tr>
<tr>
<td>Password-Valid flag is set to no</td>
</tr>
<tr>
<td>Account Expiration date is set to less than the current time plus one hour</td>
</tr>
<tr>
<td>Client flag is set to no</td>
</tr>
<tr>
<td>Account-Valid flag is set to no</td>
</tr>
<tr>
<td>Good Since date is set to greater than the current time</td>
</tr>
<tr>
<td>Password Expiration date is set to less than current time</td>
</tr>
<tr>
<td>plus one hour</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 18. Registry Policy Changes Made by the Security Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If the Security Server finds</strong></td>
</tr>
<tr>
<td>Account Lifespan is set to less than the difference between the locksmith account creation date and the current time plus 1 hour</td>
</tr>
<tr>
<td>Password Expiration date is set to greater than the time the password was last changed but less than the current time plus 1 hour</td>
</tr>
</tbody>
</table>

Use the `remote` option to allow the locksmith principal to log in from a remote machine.

Examples

All of the commands shown in the following examples must be run by root:

1. To start a Security server after you created the database with `sec_create_db`, enter:
   ```
   /opt/dcelocal/bin/secd &
   ```
2. To restart an existing replica (master or slave), enter:
   ```
   /opt/dcelocal/bin/secd &
   ```
3. To start the Security server in locksmith mode and allow the `master_admin` principal to log in on a remote machine, enter:
   ```
   /opt/dcelocal/bin/secd -locksmith master_admin -rem &
   ```
4. To end a session in locksmith mode, enter:
   ```
   kill -15 %1
   ```
sec_export_db

Starts the registry database export utility.

Format

```
sec_export_db filename [-v [verbose]]
```

Options

**filename**

Specifies the name of the export file that is created by the `sec_export_db` command. This file may be an HFS file or a sequential data set.

If a sequential data set is used, you must specify the `filename` parameter in the form:

```
//dsname or //'fully-qualified-dsname'
```

The data set must already exist and may be on any device that is supported by the QSAM access method. The record format may be fixed-length, variable-length, or undefined, and the records may be blocked or unblocked. The record length and block size may be any values that are accepted by the operating system.

**-verbose**

Causes the `sec_export_db` command to send informational messages as each registry object is written to the export file.

Usage

The `sec_export_db` command creates a portable version of the registry database. It must be issued on the system that is running the master security server for the cell. The security server must be stopped or it must be in maintenance mode. You can use the `dcecp registry disable` command to place the security server in maintenance mode, run the `sec_export_db` command, and then use the `dcecp registry enable` command to place the security server back in service.

Examples

1. The following example exports the registry to an HFS file named `/tmp/registry_export`:
   
   ```
   sec_export_db /tmp/registry_export
   ```

2. The following example exports the registry to a sequential data set named `SYS4.DCE.REGISTRY.EXPORT`:
   
   ```
   sec_export_db //'SYS4.DCE.REGISTRY.EXPORT'
   ```
sec_import_db

Starts the registry database import utility.

Format

sec_import_db filename [-replace] [-verbose]

Options

filename Specifies the name of the export file that is created by the sec_import_db command. This file may be an HFS file or a sequential data set. If a sequential data set is used, you must specify the filename parameter in the form: 

//dsname or //'fully-qualified-dsname'

The record format may be fixed-length, variable-length, or undefined, and the records may be blocked or unblocked. The record length and block size may be any values that are accepted by the operating system.

-replace Causes the existing registry database to be replaced by the registry database contained in the export file. If this option is not specified, the registry database contained in the export file is merged with the existing registry database.

-verbose Causes the sec_import_db command to issue informational messages as each registry object is imported from the export file.

Usage

The sec_import_db command imports the registry database from an export file created by the sec_export_db command. It must be issued on the system that is running the master security server for the cell. The security server must be stopped and the local registry database must exist. Use the sec_create_db command to create the initial registry database if one does not already exist.

Use the sec_import_db command to merge an exported registry after a cell reconfiguration is performed. This lets you restore all of the registry objects that existed before the cell was reconfigured.

Use the sec_import_db command to replace the registry when the registry is being recreated in order to recover from a damaged registry, or when moving from one registry database backing store to another (such as moving from an HFS registry to a DB2 registry).

The registry creator principal is used when creating new objects in the registry and the object created and changed timestamps are set to the current date and time.

Merging the Export File Into an Existing Registry: When merging an exported registry with the local registry, existing objects will not be replaced and new objects will be added. The new objects added to the registry retain their original identifications (name, UUID, and unix number). A new object will not be added to the registry if its name, UUID, or unix number is already in use.

An extended registry attribute instance is imported as long as the attribute instance does not already exist or the attribute schema entry allows multiple attribute instances.

The registry authentication policies are replaced with the values from the exported registry. The registry properties are not replaced with the values from the export registry.
The replica list is not updated with entries from the exported registry. The replica security servers that were defined in the exported registry do not accept the new security server as the master security server since it does not have the same identity as the original master security server. Any replicas that are already in the replica list are marked for initialization so that the registry changes are propagated to them when the master security server is started again.

Intercell definitions (krbtgt/cell_name) are not imported since the mirror image passwords are no longer correct.

After the merge completes, you need to configure any additional security server replicas and re-establish any intercell connections using the normal configuration methods.

The cell name and the registry creator can be different when updating an existing registry. The values from the current registry will be used.

Replacing an Existing Registry Using an Export File: When replacing the local registry, all existing objects are removed and all objects in the exported registry are added to the local registry. The replica list is imported from the export file and all of the replicas are marked for initialization. All intercell definitions are imported from the export file.

The cell name and the registry creator cannot be changed when replacing the registry. The values from the exported registry are used.

Examples

1. The following example merges the current registry with the exported registry from the HFS file named /tmp/registry_export:
   sec_import_db /tmp/registry_export

2. The following example replaces the current registry with the exported registry from the sequential data set named SYS4.DCE.REGISTRY.EXPORT:
   sec_import_db //SYS4.DCE.REGISTRY.EXPORT" -replace
storepw

Saves a z/OS user's current DCE password in the RACF database for use with single sign-on.

Format

storepw [[ password] [password]] [-r [current_password]]

Parameters

password
The DCE password associated with the DCE principal in the user's RACF DCE segment. This password is also stored in the DCE registry if the -r option is specified. If the user starts the storepw command without providing a password, storepw prompts the user for the DCE password. After receiving the password once, whether specified on the command line or because the user was prompted, the password must be specified again on the command line or the user will be prompted for the same password. The passwords supplied must be the same, or storepw fails. The user's DCE password is stored encrypted in the z/OS user's RACF DCE segment.

If the DCE password contains any blanks, storepw must be started without providing a password. Wait for the command prompt, then supply the password (including the blanks).

-r
Update the principal's DCE password in both the DCE registry and the RACF database. The default for the storepw command is to only store a user's password in the RACF database.

If this is the last parameter specified, you are prompted for the current password. The current password is not displayed when entered following a system prompt. Input following the -r option is used as the current password.

current_password
The current password establishes a DCE security context. This context is used when starting the DCE application programming interface to change the user's DCE password in the DCE registry.

If the current password contains any blanks, storepw must be started without the current password following the -r option.

The current password is not displayed when entered following a system prompt.

Usage

The storepw command is used when a user needs to change the DCE password in their RACF DCE segment. This command requires that the changed password is entered twice. This command may also be used to update your password in the DCE registry.

The user may provide none, one, or both occurrences of the password on the command line. If none is supplied, the system prompts for the changed password; if not already supplied, the system prompts for the changed password again. These passwords are verified to match. If the -r parameter is specified and the current password is not supplied, the system prompts for the current password. No password can be supplied on command invocation if it contains blanks.

If a new password is created that contains one or more blanks, subsequent uses of rgy_edit, dce_login, and storepw must be used with the prompt for the password. For example, after the -r parameter on storepw, do not specify the password when issuing the command; wait for the prompt.
If you are saving your DCE password in RACF only, (the -r option was not specified) the `storepw` command does not verify that the password you entered is the same password that is known to the DCE registry. If you enter an incorrect password twice, the password is saved. However, when you use a DCE application after that, single sign-on processing fails, and this may cause the application to fail also. If this happens, use `storepw` again, supplying the correct password.

To change your password in DCE, your current DCE password cannot be expired.

The `storepw` command can be run from either the TSO/E environment or the z/OS UNIX System Services shell.

The `storepw` command is not necessary when the DCE Security Server is running on the same system as the client. In this situation, the DCE Security Server will allow the single sign-on process to complete without requiring the DCE password to be stored in RACF.

### Examples

1. In this example, `mypw` is the current DCE password for this RACF user's DCE principal:
   ```
   storepw mypw mypw
   ```

2. In this example, the user wants to change the present DCE password and save the DCE password in both the DCE registry and the RACF database. The principal's current DCE password, which has not expired, is `alpha1x`, and the principal's new password is `beta22x5`.
   ```
   storepw beta22x5 beta22x5 -r alpha1x
   ```

3. This is the same as Example 2, however, the user is prompted for the new password.
   ```
   TEST1:/u/test1: >storepw -r alpha1x
   Enter Password: beta22x5
   Re-enter Password: beta22x5
   EUVS276521 The storepw command was successful.
   EUVS276xxI Your DCE password has been updated in the MVS manager and the DCE registry.
   ```

### Related Information:

- **Commands:**
  - `account`
  - `rgy_edit`

- **Books:**
  - [z/OS DCE Administration Guide](#)
  - [z/OS DCE User's Guide](#)
Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

<table>
<thead>
<tr>
<th>IBM</th>
<th>AnyNet</th>
<th>BookManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>CICS/ESA</td>
<td>DB2</td>
</tr>
<tr>
<td>IBMLink</td>
<td>IMS</td>
<td>IMS/ESA</td>
</tr>
<tr>
<td>Library Reader</td>
<td>MVS/ESA</td>
<td>OpenEdition</td>
</tr>
<tr>
<td>OS/390</td>
<td>Parallel Sysplex</td>
<td>RACF</td>
</tr>
<tr>
<td>Resource Link</td>
<td>SecureWay</td>
<td>VTAM</td>
</tr>
<tr>
<td>z/OS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
This glossary defines technical terms and abbreviations used in z/OS DCE documentation. If you do not find the term you are looking for, refer to the index of the appropriate z/OS DCE manual or view the IBM Glossary of Computing Terms, located at:

http://www.ibm.com/ibm/terminology

This glossary includes terms and definitions from:

- Information Technology—Portable Operating System Interface (POSIX), from the POSIX series of standards for applications and user interfaces to open systems, copyrighted by the Institute of Electrical and Electronics Engineers (IEEE).
- Information Technology Vocabulary, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1.SC1).
- Open Software Foundation (OSF).

The following abbreviations indicate terms that are related to a particular DCE service:

- **CDS**: Cell Directory Service
- **CICS/ESA®**: Customer Information Control System/ESA
- **DTS**: Distributed Time Service
- **GDS**: Global Directory Service
- **IMS/ESA®**: Information Management System/ESA
- **RPC**: Remote Procedure Call
- **Security**: Security Service
- **Threads**: Threads Service
- **XDS**: X/Open Directory Services
- **XOM**: X/Open OSI-Abstract-Data Manipulation

A

**absolute time.** A point on a time scale.

**access control list (ACL).** (1) GDS: Specifies the users with their access rights to an object. (2) Security: Data that controls access to a protected object. An ACL specifies the privilege attributes needed to access the object and the permissions that may be granted, to the protected object, to principals that possess such privilege attributes.

**access right.** Synonym for **permission**.

**accessible.** Pertaining to an object whose client possesses a valid designer or handle.

**account.** Data in the Registry database that allows a principal to log in. An account is a registry object that relates to a principal.

**ACF.** Attribute configuration file.

**ACL.** Access control list.

**adapter.** Synonym for **attachment facility**.

**aename.** An option used in Workload Balancing commands. A string, up to 18 bytes in length, referring to Application Environment.

**address.** An unambiguous name, label, or number that identifies the location of a particular entity or service. See **presentation address**.

**alias.** Synonym for **alias name**.

**alias name.** (1) GDS: A name for a directory object that consists of one or more alias entries in the directory information tree (DIT). (2) Security: An optional alternate for a principal’s primary name. Synonymous with **alias**. The alias shares the same UUID with the primary name.

**API.** Application program interface.

**application program interface (API).** A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

**architecture.** (1) The organizational structure of a computer system, including the interrelationships among
its hardware and software. (2) The logical structure and operating principles of a computer network. The operating principles of a network include those of services, functions, and protocols.

**association (connection-oriented).** A connection between a client and a server.

**attachment facility.** Application Support Server: Refers to the CICS® adapter and the IMS™ adapter. Synonymous with adapter.

**attribute.** (1) RPC: An Interface Definition Language (IDL) or attribute configuration file (ACF) that conveys information about an interface, type, field, parameter, or operation. (2) DTS: A qualifier used with DTS commands. DTS has four attribute categories: characteristics, counters, identifiers, and status. (3) XDS: Information of a particular type concerning an object and appearing in an entry that describes the object in the directory information base (DIB). It denotes the attribute’s type and a sequence of one or more attribute values, each accompanied by an integer denoting the value’s syntax.

**attribute configuration file (ACF).** RPC: An optional companion to an interface definition file that changes how the Interface Definition Language (IDL) compiler locally interprets the interface definition. See also interface definition and Interface Definition Language.

**attribute syntax.** GDS: A definition of the set of values that an attribute may assume. Attribute syntax includes the data type, in ASN.1, and usually one or more matching rules by which values may be compared.

**attribute type.** (1) XDS: The component of an attribute that indicates the type of information given by that attribute. Because it is an object identifier, it is unique among other attribute types. (2) XOM: Any of various categories into which the client dynamically groups values on the basis of their semantics. It is an integer unique only within the package.

**attribute value.** XDS, XOM: A particular instance of the type of information indicated by an attribute type.

**authentication.** In computer security, a method used to verify the identity of a principal.

**authentication level.** Synonym for protection level.

**Authentication Service.** One of three services provided by the Security Service: it verifies principals according to a specified authentication protocol. The other Security services are the Privilege Service and the Registry Service.

**authorization.** (1) The determination of a principal’s permissions with respect to a protected object. (2) The approval of a permission sought by a principal with respect to a protected object.

**authorization service.** RPC: An implementation of an authorization protocol.

**B**

**binding.** RPC: A relationship between a client and a server involved in a remote procedure call.

**binding handle.** RPC: A reference to a binding. See binding information.

**binding information.** RPC: Information about one or more potential bindings, including an RPC protocol sequence, a network address, an endpoint, at least one transfer syntax, and an RPC protocol version number. See binding. See also endpoint, network address, RPC protocol, RPC protocol sequence, and transfer syntax.

**Boolean.** Boolean algebra. The type of an expression with two binary values, “true” and “false”. Also, a variable of Boolean type or a function with Boolean arguments or result. The most common Boolean functions are AND, OR, and NOT.

In DCE Workload Balancing, when a server is workload balanced, its activate bit takes on a value of 1 (TRUE); when not balanced its activate bit takes on a value of 0 (FALSE).

**broadcast.** A notification sent to all members within an arbitrary grouping such as nodes in a network or threads in a process. See also signal.

**Browser.** CDS: A Motif-based program that lets users view the contents and structure of a cell name space.

**C**

**C interface.** The interface that is defined at a level that depends on the variant of C standardized by ANSI.

**cache.** (1) CDS: The information that a CDS clerk stores locally to optimize name lookups. The cache contains attribute values resulting from previous lookups, as well as information about other clearinghouses and namespaces. (2) Security: Contains the credentials of a principal after the DCE login. (3) GDS: See DUA cache.

**CDS.** Cell Directory Service.

**CDS clerk.** The software that provides an interface between client applications and CDS servers.

**CDS control program (CDSCP).** A command interface that CDS administrators use to control CDS
servers and clerks and manage the name space and its contents. See also manager.

**CDSCP.** CDS control program.

**cell.** The basic unit of operation in the distributed computing environment. A cell is a group of users, systems, and resources that are grouped around a common purpose and that share common DCE services.

**Cell Directory Service (CDS).** A DCE component. A distributed replicated database service that stores names and attributes of resources located in a cell. CDS manages a database of information about the resources in a group of machines called a DCE cell.

**cell-relative name.** Synonym for local name.

**characteristic attribute.** A type of attribute that reflects or affects the behavior of a software entity. Characteristic attributes can generally be set or changed.

**child directory.** CDS: A CDS directory that has a directory immediately above it is considered a child of that directory.

**child pointer.** CDS: A pointer that connects a directory to a directory immediately below it in a name space. You do not explicitly create child pointers; CDS creates them for you when you create a new directory. CDS stores the child pointer in the directory that is the parent of the new directory.

**child process.** A process, created by a parent process, that shares the resources of the parent process to carry out a request. Contrast with parent process. See also fork.

**CICS.** Customer Information Control System.

**class.** A category into which objects are placed on the basis of their purpose and internal structure.

**clearinghouse.** CDS: A collection of directory replicas on one CDS server. A clearinghouse takes the form of a database file. It can exist only on a CDS server node; it cannot exist on a node running only CDS clerk software. Usually only one clearinghouse exists on a server node.

**clearinghouse object entry.** CDS: A special class of object entry that describes a clearinghouse. The clearinghouse object entry is a pointer to the network address of an actual clearinghouse. This pointer enables CDS to find a clearinghouse and use and manage its contents. A clearinghouse changes and manages its own object entry when necessary. The clearinghouse object entry has the same name as the clearinghouse it describes.

**clerk.** (1) DTS: A software component that synchronizes the clock for its client system by requesting time values from servers, calculating a new time from the values, and supplying the computed time to client applications. (2) CDS: A software component that receives CDS requests from a client application, ascertains an appropriate CDS server to process the requests, and returns the results of the requests to the client application.

**client.** A computer or process that accesses the data, services, or resources of another computer or process on the network. Contrast with server.

**client binding information.** Information about a calling client provided by the client runtime to the server runtime, including the address where the call originated, the RPC protocol used for the call, the requested object UUID, and client authentication information. See binding information and server binding information.

**client context.** RPC: The state within an RPC server generated by a set of remote procedures and maintained across a series of calls for a particular client. See context handle. See also manager.

**client stub.** RPC: The surrogate code for an RPC interface that is linked with and called by the client application code. In addition to general operations such as marshalling data, a client stub calls the RPC runtime to perform remote procedure calls and, optionally, to manage bindings. See server stub.

**clock.** The combined hardware interrupt timer and software register that maintains the system time.

**clock adjustment.** DTS: The DTS process of changing the system clock time by changing the incremental value that is added to the clock’s software register for a specified duration.

**code page.** (1) A table showing codes assigned to character sets. (2) An assignment of graphic characters and control function meanings to all code points. (3) Arrays of code points representing characters that establish numeric order of characters. [OSF] (4) A particular assignment of hexadecimal identifiers to graphic elements. (5) Synonymous with code set. (6) See also code point, extended character.

**code set.** Synonym for code page.

**collapse.** CDS: To remove the contents of a directory from the display (close it) using the CDS Browser. To collapse an open directory, double-click on its icon. Double-clicking on a closed directory expands it. Contrast with expand.

**compatible server.** RPC: A server that offers the requested RPC interface and RPC object and that is
accessible over a valid combination of network and transport protocols. It is supported by both the client and server RPC run times.

**computed time.** DTS: The resulting time after a DTS clock synchronization. The time value that the clerk or server process computes according to the values it receives from several servers.

**context handle.** RPC: A reference to state (client context) maintained across remote procedure calls by a server on behalf of a client. See client context.

**control access.** CDS: An access right that grants users the ability to change the access control on a name and to perform other powerful management tasks, such as replicate a directory or move a clearinghouse.

**control task.** The parent process of the DCE daemons in the DCEKERN address space. All requests to start or stop DCE daemons are handled by the Control Task.

**convergence.** CDS: The degree to which CDS attempts to keep all replicas of a directory consistent. Two factors control the persistence and speed at which CDS keeps directory replicas up to date: the setting of a directory’s CDS Convergence attribute (high, medium, or low) and the background skulk time. By default, every directory inherits the convergence setting of its parent.

**conversation key.** Synonym for session key.

**copy.** GDS, XDS: Either a copy of an entry stored in other DSAs through bilateral agreement or a locally and dynamically stored copy of an entry resulting from a request (a cache copy).

**courier.** DTS: A local server that requests a time value from a randomly selected global server. The time value returned is used for synchronization.

**creation timestamp (CTS).** An attribute of all CDS clearinghouses, directories, soft links, child pointers, and object entries that contains a unique value reflecting the date and time the name was created. The timestamp consists of two parts; a time portion and a portion containing the system identifier of the node on which the name was created. These two parts guarantee uniqueness among timestamps generated on different nodes.

**credentials.** Security: A general term for privilege attribute data that has been certified by a trusted privilege certification authority.

**cross-linking information.** In order for z/OS DCE to provide RACF-DCE interoperability and single sign-on to DCE, DCE provides utilities (see mvsexpt and mvsimpt) to incorporate into RACF the information that associates a z/OS-RACF user ID with a DCE principal's identifying information and the DCE principal's UUID with the corresponding z/OS-RACF user ID. The information is placed in a RACF DCE segment and the RACF general resource class, DCEUUIDS. This is called cross-linking information and is what allows interoperability and single sign-on to work. See also interoperability and single sign-on.

**CTS.** Creation timestamp.

**Customer Information Control System (CICS).** An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

**D**

**daemon.** (1) A long-lived process that runs unattended to perform continuous or periodic system-wide functions such as network control. Some daemons are triggered automatically to perform their task; others operate periodically. An example is the cron daemon, which periodically performs the tasks listed in the crontab file. Many standard dictionaries accept the spelling demon. (2) A DCE server process.

**daemon configuration file.** A file containing information on which daemons are configured on the host, which environment variables to set, the parameters to pass to the process, minimum restart interval, and the time-out period.

**Data Encryption Standard (DES).** The National Institute of Standards and Technology (NIST) Data Encryption Standard, adopted by the U.S. government as Federal Information Processing Standard (FIPS) Publication 46, which allows only hardware implementations of the data encryption algorithm.

**datagram.** RPC: A network data packet that is independent of all other packets and does not guarantee delivery or sequentially.

**datagram protocol.** RPC: A datagram-based transport protocol, such as User Datagram Protocol (UDP), that runs over a connectionless transport protocol.

**DCE.** Distributed Computing Environment.

**DCECONF.** program used to configure and start the DCE daemons.

**DCEKERN.** The address space that contains the DCE daemons.

**default element.** RPC: An optional profile element that contains a nil interface identifier and object UUID and that specifies a default profile. Each profile can contain
only one default element. See default profile, profile, and profile element.

default profile. RPC: A backup profile referred to by the default element in another profile. The NSI import and lookup operations use the default profile, if present, whenever a search based on the current profile fails to find any useful binding information. See default element and profile.

DES. Data Encryption Standard.

descriptor. (1) XOM: The means by which the client and service exchange an attribute value and the integers that denote its representation, type, and syntax. (2) XDS: A defined data structure that represents an OM attribute type and a single value.

DFS. Distributed File Service.
directory. (1) A logical unit for storing entries under one name (the directory name) in a CDS namespace. Each physical instance of a directory is called a replica. (2) A collection of open systems that cooperates to hold a logical database of information about a set of objects in the real world.
directory ID. Directory identifier.
directory information tree (DIT). GDS: The directory information base (DIB) considered as a tree, whose vertices (other than the root) are the directory entries.
directory schema. GDS: The set of rules and constraints concerning directory information tree (DIT) structure, object class definitions, attribute types, and syntaxes that characterize the directory information base (DIB).

directory system agent (DSA). GDS: An open systems interconnection (OSI) application process that is part of the directory.
directory user agent (DUA). GDS: An open systems interconnection (OSI) application process that represents a user accessing the directory.
distinguished name (DN). GDS: One of the names of an object, formed from the sequence of RDNs of its object entry and each of its superior entries.
distributed computing. A type of computing that allows computers with different hardware and software to be combined on a network, to function as a single computer, and to share the task of processing application programs.

Distributed Computing Environment (DCE). A comprehensive, integrated set of services that supports the development, use, and maintenance of distributed applications. DCE is independent of the operating system and network; it provides interoperability and portability across heterogeneous platforms.

Distributed File Service (DFS). A DCE component. DFS joins the local file systems of several file server machines making the files equally available to all DFS client machines. DFS allows users to access and share files stored on a file server anywhere in the network, without having to consider the physical location of the file. Files are part of a single, global name space, so that a user can be found anywhere in the network by means of the same name.

Distributed Time Service (DTS). A DCE component. It provides a way to synchronize the times on different hosts in a distributed system.

DIT. Directory information tree.

DN. Distinguished name.

DNS. Domain Name System.

Domain Name System (DNS). A hierarchical scheme for giving meaningful names to hosts in a TCP/IP network.
domain name. A unique network name that is associated with a network’s unique address.
drift. DTS: The change in a clock’s error rate over a specified period of time.

DSA. Directory system agent.

DTS. Distributed Time Service.

DTS entity. DTS: The server or clerk software on a system.

DUA. Directory user agent.

DUA cache. GDS: The part of the DUA that stores information to optimize name lookups. Each cache contains copies of recently accessed object entries as well as information about DSAs in the directory.
dynamic endpoint. RPC: An endpoint that is generated by the RPC runtime for an RPC server when the server registers its protocol sequences. It expires when the server stops running. See endpoint and well-known endpoint.
**Effective Permissions.** Security: The permissions granted to a principal as a result of a masking operation.

**Element.** RPC: Any of the bits of a bit string, the octets of an octet string, or the octets by means of which the characters of a character string are represented.

**Encrypt.** To systematically encode data so that it cannot be read without knowing the coding key.

**Encryption Key.** A value used to encrypt data so that only possessors of the encryption key can decipher it.

**Endpoint.** RPC: An address of a specific server instance on a host.

**Endpoint Map.** RPC: A database local to a node where local RPC servers register binding information associated with their interface identifiers and object identifiers. The endpoint map is maintained by the endpoint map service of the DCE daemon.

**Endpoint Map Service.** RPC: A service that maintains a system's endpoint map for local RPC servers. When an RPC client makes a remote procedure call using a partially bound binding handle, the endpoint map service looks up the endpoint of a compatible local server. See **endpoint map**.

**Entity.** (1) CDS: Any manageable element through the CDS namespace. Manageable elements include directories, object entries, servers, replicas, and clerks. The CDS control program (CDSCP) commands are based on directives targeted for specific entities. (2) DTS: See **DTS Entity**.

**Entry.** GDS, XDS: The part of the DIB that contains information relating to a single directory object. Each entry consists of directory attributes.

**Entry Point Vector (EPV).** RPC: A list of addresses for the entry points of a set of remote procedures that starts the operations declared in an interface definition. The addresses are listed in the same order as the corresponding operation declarations.

**ENV.** environment variable

**Environment Variable (ENV).** A variable included in the current software environment that is available to any called program that requests it.

**Epoch Number.** DTS: An attribute that a server appends to the time values it sends to other servers.

Servers use time values only from other servers with whom they share epoch numbers.

**EPV.** Entry point vector.

**Error Tolerance.** DTS: The amount of system clock inaccuracy to which the DCE Time Service responds by abruptly setting the system clock to the computed time, rather than gradually adjusting the clock.

**Exception.** (1) An abnormal condition such as an I/O error encountered in processing a data set or a file. (2) One of five types of errors that can occur during a floating-point exception. These are valid operation, overflow, underflow, division by zero, and inexact results. [OSF] (3) Contrast with **interrupt**, **signal**.

**Expand.** CDS: To display the contents of (open) a directory using the CDS Browser. A directory that is closed can be expanded by double-clicking on its icon. Double-clicking on an expanded directory collapses it. Contrast with **collapse**.

**Export.** (1) RPC: To place the server binding information associated with an RPC interface or a list of object UUIDs or both into an entry in a name service database. (2) To provide access information for an RPC interface. Contrast with **unexport**.

**Fault.** RPC: An exception condition, occurring on a server, that is transmitted to a client.

**Filter.** An assertion about the presence or value of certain attributes of an entry to limit the scope of a search.

**Foreign Cell.** A cell other than the one to which the local machine belongs. A foreign cell and its binding information are stored in either GDS or the Domain Name System (DNS). The act of contacting a foreign cell is called intercell. Contrast with **local cell**.

**Fork.** To create and start a child process. Forking is similar to creating an address space and attaching. It creates a copy of the parent process, including open file descriptors.

**Full Name.** CDS: The complete specification of a CDS name, including all parent directories in the path from the cell root to the entry being named.

**Fully Bound Binding Handle.** RPC: A server binding handle that contains a complete server address including an endpoint. Contrast with **partially bound binding handle**.
**G**

GDA. Global Directory Agent.

GDS. Global Directory Service.

**Global Directory Agent (GDA).** A DCE component that makes it possible for the local CDS to access names in foreign cells. The GDA provides a connection to foreign cells through either the GDS or the Domain Name System (DNS).

**Global Directory Service (GDS).** A DCE component. A distributed replicated directory service that provides a global namespace that connects the local DCE cells into one worldwide hierarchy. DCE users can look up a name outside a local cell with GDS.

**global name.** A name that is universally meaningful and usable from anywhere in the DCE naming environment. The prefix /... indicates that a name is global.

**global server.** DTS: A server that provides its clock value to courier servers on other cells, or to DTS entities that have failed to obtain the specified number of servers locally.

**global set.** DTS: The group of global servers in a network.

**group.** (1) RPC: A name service entry that corresponds to one or more RPC servers that offer common RPC interfaces, RPC objects, or both. A group contains the names of the server entries, other groups, or both that are members of the group. See NSI group attribute. (2) Security: Data that associates a named set of principals that can be granted common access rights. See subject identifier.

**group member.** (1) RPC: A name service entry whose name occurs in the group. (2) Security: A principal whose name appears in a security group. See group.

**H**

**handle.** RPC: An opaque reference to information. See binding handle, context handle, interface handle, name service handle, and thread handle.

**high convergence.** CDS: A setting that controls the degree to which CDS attempts to keep all replicas of a directory consistent. High convergence means CDS makes one attempt to immediately propagate an update to all replicas. If that attempt fails (for example, if one of the replicas is unavailable), the software schedules a skulk for within one hour. Usually, a skulk occurs at least once every twelve hours on a directory with high convergence. Setting a directory’s CDS_Convergence attribute controls convergence. See low convergence and medium convergence.

**home cell.** Synonym for local cell.

**host ID.** Synonym for network address.

**identity mapping.** Application Support Server: A record in the Security Registry that contains the mapping between a client’s DCE identity and a user ID.

**IDL.** Interface Definition Language.

**IDL compiler.** RPC: A compiler that processes an RPC interface definition and an optional attribute configuration file (ACF) to generate client and server stubs, and header files. See Interface Definition Language.

**import.** (1) RPC: To obtain binding information from a name service database about a server that offers a given RPC interface by calling the RPC NSI import operation. (2) RPC: To incorporate constant, type, and import declarations from one RPC interface definition into another RPC interface definition by means of the IDL import statement.

**IMS.** Information Management System.

**inaccuracy.** DTS: The bounded uncertainty of a clock value as compared to a standard reference.

**Information Management System (IMS).** A database and data communication system capable of managing complex databases and networks in virtual storage.

**instance.** XOM: An object in the category represented by a class.

**integrity.** RPC: A protection level that may be specified in secure RPC communications to ensure that data transferred between two principals has not been changed in transit.

**interface.** RPC: A shared boundary between two or more functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. See RPC interface.

**interface definition.** RPC: A description of an RPC interface written in the DCE Interface Definition Language (IDL). See RPC interface.
Interface Definition Language (IDL). A high-level declarative language that provides syntax for interface definitions.

**interface handle.** RPC: A reference in code to an interface specification. See binding handle and interface specification.

**interface identifier.** RPC: A string containing the interface Universal Unique Identifier (UUID) and major and minor version numbers of a given RPC interface. See RPC interface.

**interface specification.** RPC: An opaque data structure that is generated by the DCE IDL compiler from an interface definition. It contains identifying and descriptive information about an RPC interface. See interface definition, interface handle, and RPC interface.

**interface UUID.** RPC: The Universal Unique Identifier (UUID) generated for an RPC interface definition using the UUID generator. See interface definition and RPC interface.

**International Organization for Standardization (ISO).** An international body composed of the national standards organizations of 89 countries. ISO issues standards on a vast number of goods and services including networking software.

**Internet address.** The 32-bit address assigned to hosts in a TCP/IP network.

**Internet Protocol (IP).** In TCP/IP, a protocol that routes data from its source to its destination in an Internet environment. IP provides the interface from the higher level host-to-host protocols to the local network protocols. Addressing at this level is usually from host to host.

**interoperability.** The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units.

**interval.** DTS: The combination of a time value and the inaccuracy associated with it; the range of values represented by a combined time and inaccuracy notation. As an example, the interval 08:00.00I00:05:00 (eight o’clock, plus or minus five minutes) contains the time 07:57.00.

**IP.** Internet Protocol

**ISO.** International Organization for Standardization

**J**

**junction.** A specialized entry in the DCE namespace that contains binding information to enable communications between different DCE services.

**K**

**Kerberos.** The authentication protocol used to carry out DCE private key authentication. Kerberos was developed at the Massachusetts Institute of Technology.

**key.** A value used to encrypt and decrypt data.

**key file.** A file that contains encryption keys for noninteractive principals.

**L**

**LAN.** Local area network.

**LFS.** local file system

**local.** (1) Pertaining to a device directly connected to a system without the use of a communication line. (2) Pertaining to devices that have a direct, physical connection. Contrast with remote.

**local area network (LAN).** A network in which communication is limited to a moderate-sized geographical area (1 to 10 km) such as a single office building, warehouse, or campus, and which does not generally extend across public rights-of-way. A local network depends on a communication medium capable of moderate to high data rate (greater than 1Mbps), and normally operates with a consistently low error rate.

**local cell.** The cell to which the local machine belongs. Synonymous with home cell. Contrast with foreign cell.

**locale.** Computer locales define the user's environment -- the conventions for a specific language and culture, including appropriate date and time formatting, character conversion, sorting and text handling. These locales are collections of processing variables used to specify how a process will run.

**local file system (LFS).** An organized collection of data in the form of a root directory and its subdirectories and files. An LFS supports special features useful in a distributed environment: the ability to replicate data; to log file system data, enabling quick recovery after a system malfunction; to simplify administration by dividing the file system into easily managed units called files; and to associate access control lists (ACLs) with files and directories. An LFS is located on a disk that is physically attached to a machine in other file
systems, a single disk partition contains only one file system. In DCE LFS an aggregate can contain multiple file systems (filesets). See also access control list (ACL).

local name. A name that is meaningful and usable only within the cell where an entry exists. The local name is a shortened form of a global name. Local names begin with the prefix /: and do not contain a cell name. Synonymous with cell-relative name.

local server. DTS: A server that synchronizes with its peers and provides its clock value to other servers and clerks in the same network.

local set. DTS: A collection of the servers in a particular network.

low convergence. A setting that controls the degree to which CDS attempts to keep all replicas of a directory consistent. Low convergence means CDS does not immediately propagate an update; it simply waits for the next skulk to distribute all updates that occurred since the last skulk. Skulks occur at least once every 24 hours on directories with low convergence. Low convergence helps conserve resources by avoiding update propagations between skulks. Setting a directory's CDS_Convergence attribute controls convergence. See high convergence and medium convergence.

M

manager. RPC: A set of remote procedures that implement the operations of an RPC interface and that can be dedicated to a given type of object. See also object and RPC interface.

manager entry point vector. RPC: The runtime code on the server side uses this entry point vector to dispatch incoming remote procedure calls. See entry point vector and manager.

marshalling. RPC: The process by which a stub converts local arguments into network data and packages the network data for transmission. Contrast with unmarshalling.

mask. (1) A pattern of characters used to control the retention or deletion of portions of another pattern of characters (2) Security: Used to establish maximum permissions that can then be applied to individual ACL entries. (3) GDS: The administration screen interface menus.

master replica. CDS: The first instance of a specific directory in the namespace. After copies of the directory have been made, a different replica can be designated as the master, but only one master replica of a directory can exist at a time. CDS can create, update, and delete object entries and soft links in a master replica.

medium convergence. CDS: A setting that controls the degree to which CDS attempts to keep all replicas of a directory consistent. Medium convergence means CDS makes one attempt to immediately propagate an update to all replicas of the directory in which a change was made. If the attempt fails, the software lets the next scheduled skulk make the replicas consistent. Skulks occur at least once every 12 hours on a directory with medium convergence. When a name space is created, the default setting on the root directory is medium. Setting a directory's CDS_Convergence attribute controls convergence. See high convergence and low convergence.

MODIFY DCEKERN. MODIFY command used to start, stop, and display the status of DCE daemons.

mutex. Mutual exclusion. A read/write lock that grants access to only a single thread at any one time. A mutex is often used to ensure that shared variables are always seen by other threads in a consistent way.

mvsexpt. One of two (the other is mvsimpt) utilities used to automate much of the administrator's work in creating the cross-linking information for DCE-RACF interoperability. The mvsexpt utility creates the cross-linking information in the RACF database from information in the DCE registry. See also cross-linking information, interoperability, and single sign-on.

mvsimpt. One of two (the other is mvsexpt) utilities used to automate much of the administrator's work in creating the cross-linking information for DCE-RACF interoperability. The mvsimpt utility creates DCE principals from information obtained from the RACF database. See also cross-linking information, interoperability, and single sign-on.

N

name. GDS, CDS: A construct that singles out a particular (directory) object from all other objects. A name must be unambiguous (denote only one object); however, it need not be unique (be the only name that unambiguously denotes the object).

name service. A central repository of named resources in a distributed system. In DCE, this is the same as Directory Service.

name service handle. RPC: An opaque reference to the context used by the series of next operations called during a specific name service interface (NSI) search or inquiry.

name service interface (NSI). RPC: A part of the application program interface (API) of the RPC run time.
NSI routines access a name service, such as CDS, for RPC applications.

namespace.  CDS: A complete set of CDS names that one or more CDS servers look up, manage, and share. These names can include directories, object entries, and soft links.

NDR.  Network Data Representation.

network.  A collection of data processing products connected by communications lines for exchanging information between stations.

network address.  An address that identifies a specific host on a network. Synonymous with host ID.

network data.  RPC: Data represented in a format defined by a transfer syntax. See also transfer syntax.

Network Data Representation (NDR).  RPC: The transfer syntax defined by the Network Computing Architecture. See transfer syntax.

network protocol.  A communications protocol from the Network Layer of the Open Systems Interconnection (OSI) network architecture, such as the Internet Protocol (IP).

Network Time Protocol (NTP).  A clock synchronization protocol commonly used on an Internet.

node.  (1) An endpoint of a link, or a junction common to two or more links in a network. Nodes can be preprocessors, controllers, or workstations, and they can vary in routing and other functional capabilities. (2) In network topology, the point at an end of a branch. It is usually a physical machine.

null time provider.  The daemon that fetches the time from the hardware clock of the DCE host for DTS.

NSI.  Name service interface.

NSI binding attribute.  RPC: An RPC-defined attribute (NSI attribute) of a name service entry; the binding attribute stores binding information for one or more interface identifiers offered by an RPC server and identifies the entry as an RPC server entry. See binding information and NSI object attribute. See also server entry.

NSI group attribute.  RPC: An RPC-defined attribute (NSI attribute) of a name service entry that stores the entry names of the members of an RPC group and identifies the entry as an RPC group. See group.

NSI object attribute.  RPC: An RPC-defined attribute (NSI attribute) of a name service entry that stores the object UUIDs of a set of RPC objects. See object.

NSI profile attribute.  RPC: An RPC-defined attribute (NSI attribute) of a name service entry that stores a collection of RPC profile elements and identifies the entry as an RPC profile. See profile.

NTP.  Network Time Protocol.

NULL.  In the C language, a pointer that does not point to a data object.

O

object.  (1) A data structure that implements some feature and has an associated set of operations. (2) RPC: For RPC applications, anything that an RPC server defines and identifies to its clients using an object Universal Unique Identifier (UUID). An RPC object is often a physical computing resource such as a database, directory, device, or processor. Alternatively, an RPC object can be an abstraction that is meaningful to an application, such as a service or the location of a server. See object UUID. (3) XDS: Anything in the world of telecommunications and information processing that can be named and for which the directory information base (DIB) contains information. (4) XOM: Any of the complex information objects created, examined, changed, or destroyed by means of the interface.

object class.  GDS, CDS: An identified family of objects that share certain characteristics. An object class can be specific to one application or shared among a group of applications. An application interprets and uses an entry’s class-specific attributes based on the class of the object that the entry describes.

object class table (OCT).  A recurring attribute of the directory schema with the description of the object classes permitted.

object entry.  CDS: The name of a resource (such as a node, disk, or application) and its associated attributes, as stored by CDS. CDS administrators, client application users, or the client applications themselves can give a resource an object name. CDS supplies some attribute information (such as a creation timestamp) to become part of the object, and the client application may supply more information for CDS to store as other attributes. See entry.

object identifier (OID).  A value (distinguishable from all other such values) that is associated with an information object. It is formally defined in the CCITT X.208 standard.

object management (OM).  The creation, examination, change, and deletion of potentially complex information objects.
object name.  CDS: A name for a network resource.

object UUID.  RPC: The Universal Unique Identifier (UUID) that identifies a particular RPC object. A server specifies a distinct object UUID for each of its RPC objects. To access a particular RPC object, a client uses the object UUID to find the server that offers the object. See object.

ectal. In reference to a selection, choice or condition that has eight possible different values or states. In reference to a fixed-radix numeration having a radix of eight.

OCT. Object class table.

OID. Object identifier.

OM. Object management.

Open Software Foundation (OSF). A nonprofit research and development organization set up to encourage the development of solutions that allow computers from different vendors to work together in a true open-system computing environment.

open systems interconnection (OSI). The interconnection of open systems in accordance with standards of the International Organization for Standardization (ISO) for the exchange of information.

operation. (1) GDS: Processing performed within the directory to provide a service, such as a read operation. (2) RPC: The task performed by a routine or procedure that is requested by a remote procedure call.

organization. (1) The third field of a subject identifier. (2) Security: Data that associates a named set of users who can be granted common access rights that are usually associated with administrative policy.

OSF. Open Software Foundation.

OSI. Open systems interconnection

P

PAC. Privilege attribute certificate.

package. XOM: A specified group of related object management (OM) classes, denoted by an object identifier.

parent directory. CDS: Any directory that has one or more levels of directories beneath it in a cell name space. A directory is the parent of any directory immediately beneath it in the hierarchy.

parent process. A process created to carry out a program. The parent process in turn creates child processes to process requests. Contrast with child process.

partially bound binding handle. RPC: A server binding handle that contains an incomplete server address lacking an endpoint. Contrast with fully bound binding handle.

Partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

password. A secret string of characters shared between a computer system and a user. The user must specify the character string to gain access to the system.

PCS. Portable Character Set.

PDS. Partitioned data set

peer trust. A type of trust relationship established between two cells by means of a secret key shared by authentication surrogates maintained by the two cells. A peer trust relationship enables principals in one cell to communicate securely with principals in the other.

permission. (1) The modes of access to a protected object. The number and meaning of permissions with respect to an object are defined by the access control list (ACL) Manager of the object. (2) GDS: One of five groups that assigns modes of access to users: MODIFY PUBLIC, READ STANDARD, MODIFY STANDARD, READ SENSITIVE, or MODIFY SENSITIVE. Synonymous with access right. See also access control list.

person. See principal.

pickle. A type of data encoding. When a Remote Procedure Call (RPC) sends data between a client and a server, it serializes the user’s data structures by using the IDL Encoding Services (ES). This serialization scheme for encoding and decoding data is informally called pickling.

ping. Utility in TCP/IP that is used to test if a destination host can be reached by sending test packets and waiting for a reply.

In the RPC control program, a command to test if a server is listening.

pipe. (1) RPC: A mechanism for passing large amounts of data in a remote procedure call. (2) The data structure that represents this mechanism.

plaintext. The input to an encryption function or the output of a decryption function. Encryption transforms
plaintext to ciphertext and decryption transforms ciphertext into plaintext.

**platform.** The operating system environment in which a program runs.

**PLT.** Program list table.

**port.** (1) Part of an Internet Protocol (IP) address specifying an endpoint. (2) To make the programming changes necessary to allow a program that runs on one type of computer to run on another type of computer.

**Portable Character Set.** A set of characters to enable internationalization. A character set used by DCE to enable word wide connectivity by ensuring that a minimum group of characters is supported in DCE. All DCE RPC clients and servers are required to support the DCE PCS.

**position (within a string).** XOM: The ordinal position of one element of a string relative to another.

**position (within an attribute).** XOM: The ordinal position of one value relative to another.

**potential binding.** RPC: A specific combination of an RPC protocol sequence, RPC protocol major version, network address, endpoint, and transfer syntax that an RPC client can use to establish a binding with an RPC server. See binding. See also endpoint, network address, RPC protocol, RPC protocol sequence, and transfer syntax.

**presentation address.** An unambiguous name that identifies a set of presentation service access points. Loosely, it is the network address of an open systems interconnection (OSI) service.

**primary name.** The string name of an object to which any aliases for that object refer. The DCE refers to objects by their primary names, although DCE users may refer to them by their aliases.

**principal.** Security: An entity that can communicate securely with another entity. In the DCE, principals are represented as entries in the Registry database and include users, servers, computers, and authentication surrogates.

**privacy.** RPC: A protection level that encrypts RPC argument values. in secure RPC communications.

**privilege attribute certificate (PAC).** Security: Data describing a principal’s privilege attributes that has been certified by an authority. In the DCE, the Privilege Service is the certifying authority; it seals the privilege attribute data in a ticket. The authorization protocol, DCE Authorization, determines the permissions granted to principals by comparing the privilege attributes in PACs with entries in an access control list.

**privilege ticket.** Security: A ticket that contains the same information as a simple ticket, and also includes a privilege attribute certificate. See service ticket, simple ticket, and ticket-granting ticket.

**profile.** RPC: An entry in a name service database that contains a collection of elements from which name service interface (NSI) search operations construct search paths for the database. Each search path is composed of one or more elements that refer to name service entries corresponding to a given RPC interface and, optionally, to an object. See NSI profile attribute and profile element.

**profile element.** RPC: A record in an RPC profile that maps an RPC interface identifier to a profile member (a server entry, group, or profile in a name service database). See profile. See also group, interface identifier and server entry.

**profile member.** RPC: A name service entry whose name occupies the member field of an element of the profile. See profile.

**program list table (PLT).** CICS/ESA: A data area that contains a list of programs to be started.

**programming interface.** The supported method through which customer programs request software services. The programming interface consists of a set of callable services provided with the product.

**protection level.** The degree to which secure network communications are protected. Synonymous with authentication level.

**protocol.** A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication.

**protocol sequence.** Synonym for RPC protocol sequence.

**R**

**RACF.** Resource Access Control Facility.

**RDN.** Relative distinguished name.

**read-only replica.** (1) CDS: A copy of a CDS directory in which applications cannot make changes. Although applications can look up information (read) from it, they cannot create, change, or delete entries in a read-only replica. Read-only replicas become consistent with other, changeable replicas of the same directory during skulks and routine propagation of updates. (2) Security: A replicated Registry server.
register.  (1) RPC: To list an RPC interface with the
RPC runtime. (2) To place server-addressing
information into the local endpoint map. (3) To insert
authorization and authentication information into binding
information. See endpoint map and RPC interface.

Registry database.  Security: A database of security
information about principals, groups, organizations,
accounts, and security policies.

Registry replica.  Security: A read-only instance of a
Registry database.

Registry Service.  Security: One of three services
provided by the Security Service; the Registry Service
manages information about principals, accounts, and
security policies. The other services are the Privilege
Service and the Authentication Service.

relative distinguished name (RDN).  GDS, XDS: A
set of Attribute Value Assertions (AVAs).

relative time.  A discrete time interval that is usually
added to or subtracted from an absolute time. See
absolute time.

remote.  Pertaining to a device, file or system that is
accessed by your system through a communications
line. Contrast with local.

remote procedure.  RPC: An application procedure
located in a separate address space from calling code.
See remote procedure call.

remote procedure call.  RPC: A client request to a
service provider located anywhere in the network.

Remote Procedure Call (RPC).  A DCE component. It
allows requests from a client program to access a
procedure located anywhere in the network.

replica.  CDS: A directory in the CDS namespace.
The first instance of a directory in the name space is
the master replica. See master replica and read-only
replica.

replica set.  CDS: The set of all copies of a CDS
directory. Information about a directory’s replica set is
contained in an attribute of directories and child pointers
called CDS_Replicas. The attribute contains the type
of each replica (master or read-only) and the
clearinghouse where it is located. When skulking a
directory, CDS refers to the directory’s replica set to
ensure that it finds all copies of that directory. During a
lookup, CDS may refer to the replica set in a child
pointer when trying to locate a directory that does not
exist in the local clearinghouse.

replication.  The making of a shadow of a database to
be used by another node. Replication can improve
availability and load-sharing.

request.  A command sent to a server over a
connection.

resource.  Items such as printers, plotters, data
storage, or computer services. Each has a unique
identifier associated with it for naming purposes.

Resource Access Control Facility (RACF).  An IBM
licensed program, that provides for access control by
identifying and verifying the users to the system,
authorizing access to protected resources, and logging
the detected unauthorized access to protected
resources.

return value.  A function result that is returned in
addition to the values of any output or input/output
arguments.

ROM.  Read-only memory.

RPC.  Remote Procedure Call.

RPC control program (RPCCP).  An interactive
administrative facility for managing name service entries
and endpoint maps for RPC applications.

RPCCP.  RPC control program

RPC interface.  A logical group of operations, data
types, and constant declarations that serves as a
network contract for a client to request a procedure in a
server. See also interface definition and operation.

RPC protocol.  An RPC-specific communications
protocol that supports the semantics of the DCE RPC
API and runs over either connectionless or
connection-oriented communications protocols.

RPC protocol sequence.  A valid combination of
communications protocols represented by a character
string. Each RPC protocol sequence typically includes
three protocols: a network protocol, a transport protocol,
and an RPC protocol that works with the network and
transport protocols. See network protocol, RPC
protocol, and transfer protocol. Synonymous with
protocol sequence.

RPC runtime.  A set of operations that manages
communications, provides access to the name service
database, and performs other tasks, such as managing
servers and accessing security information, for RPC
applications. See RPC runtime library.

RPC runtime library.  A group of routines of the RPC
runtime that support the RPC applications on a system.
The runtime library provides a public interface to
application programmers, the application programming
interface (API), and a private interface to stubs, the stub
programming interface (SPI). See RPC runtime.
schema. See directory schema.

Security Service. A DCE component that provides trustworthy identification of users, secure communications, and controlled access to resources in a distributed system.

segment. One or more contiguous elements of a string.

server. (1) On a network, the computer that contains programs, data, or provides the facilities that other computers on the network can access. (2) The party that receives remote procedure calls. Contrast with client.

server binding information. RPC: Binding information for a particular RPC server. See binding information and client binding information.

server entry. RPC: A name service entry that stores the binding information associated with the RPC interfaces of a particular RPC server and object Universal Unique Identifiers (UUIDs) for any objects offered by the server. See also binding information, NSI binding attribute, NSI object attribute, object and RPC interface.

server instance. RPC: A server running in a specific address space. See server.

server state. Application Support Server: The condition of the Application Support Server after it has been started. The server state may be any of the following, depending on the actions directed to it by the administrator: initializing, quiescent, starting, operating, or stopping.

server stub. RPC: The surrogate calling code for an RPC interface that is linked with server application code containing one or more sets of remote procedures (managers) that implement the interface. See client stub. See also manager.

service. In network architecture, the capabilities that the layers closer to the physical media provide to the layers closer to the end user.

service ticket. Security: A ticket for a specified service other than the ticket-granting service. See privilege ticket, simple ticket, and ticket-granting ticket.

session. GDS: A sequence of directory operations requested by a particular user of a particular directory user agent (DUA) using the same session object management (OM) object.

session key. Security: A short-lived encryption key provided by the Authentication Service to two principals for the purpose of ensuring secure communications between them. Synonymous with conversation key.

shell script. A file containing shell commands. If the file can be processed, you can specify its name as a simple command. Processing of a shell script causes a shell to run the commands in the script. Alternatively, a shell can be requested to run the commands in a shell script by specifying the name of the shell script as the operand sh utility.

signal. Threads: To wake only one thread waiting on a condition variable. See broadcast.

signed. Security: Pertaining to information that is appended to an enciphered summary of the information. This information ensures the integrity of the data, the authenticity of the originator, and the unambiguous relationship between the originator and the data.

sign-on. (1) A procedure to be followed at a terminal or workstation to establish a link to a computer. (2) To begin a session at a workstation. (3) Same as log on or log in.

simple name. CDS: One element in a CDS full name. Simple names are separated by slashes in the full name.

simple ticket. Security: A ticket that contains the principal's identity, a session key, a timestamp and other information, sealed using the target's secret key. See privilege ticket, service ticket, and ticket-granting ticket.

single sign-on. In z/OS DCE, single sign-on to DCE allows a z/OS user who has already been authenticated to an external security manager, such as RACF, to be logged in to DCE. DCE does this automatically when a DCE application is started, if the user is not already logged in to DCE.

skew. The time difference between two clocks or clock values.

skulk. CDS: A process by which CDS makes the data consistent in all replicas of a particular directory. CDS collects all changes made to the master replica since the last skulk was completed, and disseminates the changes from the up-to-date replica to all other existing replicas of the directory. All replicas of a directory must be available for a skulk to be considered successful.

socket. A unique host identifier created by the concatenation of a port identifier with a TCP/IP address.

soft link. CDS: A pointer that provides an alternative name for an object entry, directory, or other soft link in the name space. A soft link can be permanent or it can
expire after a specific period of time. The CDS server also can delete it after the name that the link points to is deleted.

**specific.** XOM: The attribute types that can appear in an instance of a given class, but not in an instance of its superclasses.

**standard.** A model that is established and widely used.

**string.** An ordered sequence of bits, octets, or characters, accompanied by the string’s length.

**stub.** RPC: A code module specific to an RPC interface that is generated by the Network Interface Definition Language (NIDL) compiler to support remote procedure calls for the interface. RPC stubs are linked with client and server applications and hide the intricacies of remote procedure calls from the application code. See *client stub* and *server stub*.

**subject identifier (SID).** A string that identifies a user or set of users. Each SID consists of three fields in the form person.group.organization. In an account, each field must have a specific value; in an access control list (ACL) entry, one or more fields may use a wildcard.

**subordinate.** GDS, XDS: In the directory information tree (DIT), an entry whose distinguished name includes that of the other as a prefix.

**synchronization.** DTS: The process by which a Distributed Time Service entity requests clock values from other systems, computes a new time from the values, and adjusts its system clock to the new time.

**syntax.** (1) XOM: An object management (OM) syntax is any of the various categories into which the OM specification statically groups values on the basis of their form. These categories are additional to the OM type of the value. (2) A category into which an attribute value is placed on the basis of its form. See *attribute syntax*.

**system time.** The time value maintained and used by the operating system.

**T**

**TCP.** Transmission Control Protocol

**TCP/IP.** Transmission Control Protocol/Internet Protocol

**TDF.** Time differential factor.

**thread.** A single sequential flow of control within a process.

**thread handle.** RPC: A data item that enables threads to share a storage management environment.

**ticket.** Security: An application-transparent mechanism that transmits the identity of an initiating principal to its target. See *privilege ticket*, *service ticket*, *simple ticket*, and *ticket-granting ticket*.

**ticket-granting ticket.** Security: A ticket to the ticket-granting service. See *privilege ticket*, *service ticket*, and *simple ticket*.

**time differential factor (TDF).** DTS: The difference between universal time coordinated (UTC) and the time in a particular time zone.

**time provider (TP).** DTS: A process that queries universal time coordinated (UTC) from a hardware device and provides it to the server.

**time provider program.** DTS: An application that functions as a time provider.

**tower.** CDS: A set of physical address and protocol information for a particular server. CDS uses this information to locate the system on which a server resides and to determine which protocols are available at the server. Tower values are contained in the *CDS_Towers* attribute associated with the object entry that represents the server in the cell namespace.

**TP.** time provider

**transfer syntax.** RPC: A set of encoding rules used for transmitting data over a network and for converting application data to and from different local data representations. See also *Network Data Representation*.

**Transmission Control Protocol (TCP).** A communications protocol used in Internet and any other network following the U.S. Department of Defense standards for inter-network protocol. TCP provides a reliable host-to-host protocol in packet-switched communication networks and in an interconnected system of such networks. It assumes that the Internet Protocol is the underlying protocol. The protocol that provides a reliable, full-duplex, connection-oriented service for applications.

**Transmission Control Protocol/Internet Protocol (TCP/IP).** A set of non-proprietary communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

**transport protocol.** A communications protocol, such as the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP).
**type.** XOM: A category into which attribute values are placed on the basis of their purpose. See *attribute type*.

**type UUID.** RPC: The Universal Unique Identifier (UUID) that identifies a particular type of object and an associated manager. See also *manager* and *object*.

**U**

**UDP.** User Datagram Protocol.

**unexport.** RPC: To remove binding information from a server entry in a name service database. Contrast with *export*.

**Universal Time Coordinated (UTC).** The basis of standard time throughout the world. Synonymous with Greenwich mean time (GMT).

**Universal Unique Identifier (UUID).** RPC: An identifier that is immutable and unique across time and space. A UUID can uniquely identify an entity such as an object or an RPC interface. See *interface UUID*, *object UUID*, and *type UUID*.

**unmarshalling.** RPC: The process by which a stub disassembles incoming network data and converts it into local data in the appropriate local data representation. Contrast with *marshalling*.

**update timestamp (UTS).** CDS: An attribute that identifies the time at which the most recent change was made to any attribute of a particular CDS name. For directories, the UTS reflects changes made only to attributes that apply to the actual directory (not one of its replicas).

**user.** A person who requires the services of a computing system.

**User Datagram Protocol (UDP).** In TCP/IP, a packet-level protocol built directly on the Internet protocol layer. UDP is used for application-to-application programs between TCP/IP host systems.

**UTC.** Universal Time Coordinated

**UTS.** Update timestamp.

**UUID.** Universal unique identifier

**V**

**value.** XOM: An arbitrary and complex information item that can be viewed as a characteristic or property of an object. See *attribute value*.

**vector.** RPC: An array of references to other structures.

**vendor.** Supplier of software products.

**Virtual Telecommunications Access Method (VTAM®).** An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

**W**

**WAN.** Wide area network.

**WLB.** Workload Balancing

**WLM.** Workload Management

**well-known endpoint.** RPC: A preassigned, stable endpoint that a server can use every time it runs. Well-known endpoints typically are assigned by a central authority responsible for a transport protocol. An application declares a well-known endpoint either as an attribute in an RPC interface header or as a variable in the server application code. See *dynamic endpoint* and *endpoint*.

**wide area network (WAN).** A network that provides communication services to a geographic area larger than that served by a local area network (LAN).

**Workload Balancing (WLB).** The task used in a parallel sysplex subsystem (available in z/OS DCE or OS/390 DCE Release 6 or higher). Workload Balancing allows DCE servers that export the same interfaces within a Sysplex to have their work distributed evenly to them from their corresponding DCE clients.

**Workload Management (WLM).** The service available on z/OS, used in a parallel sysplex subsystem. Workload Management allows a system programmer to provide a Coupled Data Set (CDS) along with a set of service policies describing the amount of resource that the sysplex is able to supply to do work. For DCE Workload Balancing, the WLM product must be configured and the WLM subsystem must be operating in "goal mode".
X.500. The CCITT/ISO standard for the open systems interconnection (OSI) application-layer directory. It allows users to register, store, search, and retrieve information about any objects or resources in a network or distributed system.

XDS. The X/Open Directory Services API.

X/Open Directory Services (XDS). An application program interface that DCE uses to access its directory service components. XDS provides facilities for adding, deleting, and looking up names and their attributes. The XDS library detects the format of the name to be looked up and directs the calls it receives to either GDS or CDS. XDS uses the XOM API to define and manage its information.

XOM. The X/Open OSI-Abstract-Data Manipulation API.
Bibliography

This bibliography is a list of publications for z/OS DCE and other products. The complete title, order number, and a brief description is given for each publication.

z/OS DCE Publications

This section lists and provides a brief description of each publication in the z/OS DCE library.

Overview

- **z/OS DCE Introduction**, GC24-5911
  This book introduces z/OS DCE. Whether you are a system manager, technical planner, z/OS system programmer, or application programmer, it will help you understand DCE and evaluate the uses and benefits of including z/OS DCE as part of your information processing environment.

Planning

- **z/OS DCE Planning**, GC24-5913
  This book helps you plan for the organization and installation of z/OS DCE. It discusses the benefits of distributed computing in general and describes how to develop plans for a distributed system in a z/OS environment.

Administration

- **z/OS DCE Configuring and Getting Started**, SC24-5910
  This book helps system and network administrators configure z/OS DCE.

- **z/OS DCE Administration Guide**, SC24-5904
  This book helps system and network administrators understand z/OS DCE and tells how to administer it from the batch, TSO, and shell environments.

- **z/OS DCE Command Reference**, SC24-5909
  This book provides reference information for the commands that system and network administrators use to work with z/OS DCE.

- **z/OS DCE User's Guide**, SC24-5914
  This book describes how to use z/OS DCE to work with your user account, use the directory service, work with namespaces, and change access to objects that you own.

Application Development

- **z/OS DCE Application Development Guide: Introduction and Style**, SC24-5907
  This book assists you in designing, writing, compiling, linking, and running distributed applications in z/OS DCE.

- **z/OS DCE Application Development Guide: Core Components**, SC24-5905
  This book assists programmers in developing applications using application facilities, threads, remote procedure calls, distributed time service, and security service.

- **z/OS DCE Application Development Guide: Directory Services**, SC24-5906
  This book describes the z/OS DCE directory service and assists programmers in developing applications for the cell directory service and the global directory service.

- **z/OS DCE Application Development Reference**, SC24-5908
  This book explains the DCE Application Program Interfaces (APIs) that you can use to write distributed applications on z/OS DCE.

Reference

- **z/OS DCE Messages and Codes**, SC24-5912
  This book provides detailed explanations and recovery actions for the messages, status codes, and exception codes issued by z/OS DCE.

z/OS SecureWay® Security Server Publications

This section lists and provides a brief description of books in the z/OS SecureWay Security Server library that may be needed for z/OS SecureWay Security Server DCE and for RACF® interoperability.
This book describes the Lightweight Directory Access Protocol (LDAP) client APIs that you can use to write distributed applications on z/OS DCE and gives you information on how to develop LDAP applications.

This book explains RACF concepts and describes how to plan for and implement RACF.

This book describes how to install, configure, and run the LDAP server. It is intended for administrators who will maintain the server and database.

This book provides the configuration, commands, messages, examples and problem determination for the z/OS Firewall Technologies. It is intended for network or system security administrators who install, administer and use the z/OS Firewall Technologies.

This non-IBM book on the Tool Control Language is useful for application developers, DCECP script writers, and end users.

This book describes how to develop applications in the C/C++ language in z/OS.

This book helps system and network administrators understand and administer Application Support.

This book provides information on using Application Support to develop applications that can access CICS® and IMS™ transactions.
Encina Publications

- **z/OS Encina Toolkit Executive Guide and Reference** SC24-5919
  This book discusses writing Encina applications for z/OS.

- **z/OS Encina Transactional RPC Support for IMS** SC24-5920
  This book is to help software designers and programmers extend their IMS transaction applications to participate in a distributed, transactional client/server application.
Index

A
abbreviating CDS internal commands 528
abbreviations, dcecp 129
Access Control List (ACL)
   DCE login
      destroying context 650
      validating identity 648
   editing 641
   entries 645
   interactive subcommands 644
   key identifier 646
   listing 641
   locksmith mode, daemon account changes 688
   permissions 647
   registry database, exporting 690
   registry database, importing 691
   security server daemon, starting 688
   server registry database, creating 685
   type tag 645
account (user) expiration 649
account catalog command 17
account command 13
account create command 19
account delete command 21
account generate command 22
account help command 23
account modify command 24
account operations command 26
account show command 27
accounts 663
acl command 29
acl delete command 36
acl help command 37
acl modify command 38
acl operations command 40
acl permissions command 41
acl replace command 42
acl show command 44
acl_edit
   See acledit
acledit (continued)
   interactive subcommands (continued)
      exit 644
      get access 644
      help 644
      kill 644
      list 644
      modify 644
      permissions 644
      substitute 644
      test 644
      starting 641
acledit command 641
add directory command 515
add element command 438
add entry command 441
add mapping command 443
add member command 445
add object command 517
administration, security 676
advertise command 610
arguments lists, dcecp 130
attribute lists, dcecp 131
attribute options, dcecp 131
attribute types (cdscp) 525
attribute value
   adding to directory 515
   adding to object entry 517
   changing for directory 571
   changing for object entry 578
   removing from directory 564
   removing from object entry 567
attrlist add command 47
attrlist command 46
   manipulating 46
attrlist getvalues command 48
attrlist help command 49
attrlist list command 50
attrlist modify command 51
attrlist operations command 52
attrlist remove command 53
aud command 54
aud disable command 56
aud enable command 57
aud help command 58
aud modify command 59
aud permissions command 60
aud replace command 61
aud show command 62
aud stop command 63
audevents catalog command 65
audevents command 64
audevents help command 66
audevents operations command 67
audevents show command 68
audfilter catalog command 71
audfilter command 69
audfilter create command 72
audfilter delete command 73
audfilter help command 74
audfilter modify command 75
audfilter operations command 76
audfilter show command 77
audtrail command 78
audtrail help command 79
audtrail operations command 80
audtrail show command 81

B
batch file, creating in CDSCP 550
bibliography 715
binding information from server
  exporting 448
  importing 463
  showing 500
  unexporting 502
binding information, dcecp 132
books, list of DCE and related 715

C
caching Kerberos ticket-granting ticket 652
CDS administrative utilities
  ldap_addcell utility 511
  mkdcregister utility 513

cds daemon
  start options 4

CDS servers
  clearing clearinghouses 532
  stopping 549
  viewing attributes 607
cdsadv daemon
  description 508
  start options 3
cds汝cache command 82
cds汝cache create command 83
cds汝cache delete command 85
cds汝cache dump command 86
cds汝cache help command 87
cds汝cache operations command 88
cds汝cache show command 89
cdsclerk daemon
  description 508
  start options 4
cds汝 (CDS control program) (continued)
  elements 524
  entering internal commands 528
  entity names 524
  privilege required 529
  starting 518
  time offset 528
  using comment characters 529
  using wildcard characters 528
  verbs 524

cds汝 commands
  clear clearinghouse 532
  create clearinghouse 534
  delete clearinghouse 543
  disable server 549
  introduction 507
  show server 607
  supported
    add directory 515
    add object 517
    cds汝 518
    clear cached server 531
    create child 533
    create clearinghouse 534
    create directory 535
    create link 536
    create object 538
    create replica 539
    define cached server 540
    delete child 542
    delete clearinghouse 543
    delete directory 544
    delete link 545
    delete object 546
    delete replica 547
    disable clerk 548
    do 550
    dump clerk cache 551
    help 552
    ldap_addcell utility 511
    list child 555
    list clearinghouse 556
    list directory 558
    list link 560
    list object 562
    mkdcregister utility 513
    remove directory 564
    remove link 566
    remove object 567
    set cds汝 confidence 569
    set cds汝 preferred clearinghouse 570
    set directory 571
    set directory to new epoch 573
    set directory to skulk 575
    set link 576
    set object 578
    show cached clearinghouse 580
cdscp commands (continued)
supported (continued)
  show cached server  582
  show cdscp confidence  583
  show cdscp preferred clearinghouse  584
  show cdscp syntax  585
  show cell  588
  show child  590
  show clearinghouse  592
  show clerk  595
  show directory  597
  show link  600
  show object  602
  show replica  604
unsupported
  clear clearinghouse  507
  create clearinghouse  507
  delete clearinghouse  507
  disable server  507
  show server  507

cdscp syntax  585
cell command  91
dcecp object  91

Cell Directory Service (CDS)
control program (cdscp)
  attribute types and structures  525
  attributes  525
  elements  524
  entering internal commands  528
  entity names  524
  privilege required  529
  starting  518
  time offset  528
  using comment characters  529
  using wildcard characters  528
  verbs  524
daemons
  description  508
  start options  3, 4
daemons supported
cdsadv  508
cdsclerk  508
daemons unsupported
cdsd  508
gdad  508
introduction  507
cell help command  92
cell information, displaying  588
cell name  475
cell operations command  93
cell ping command  94
cell show command  96
cell-relative name  476
change command  611
child pointers
  creating  533

child pointers (continued)
deleting  542
displaying attribute information  590
listing  555
clear cached server command  531
clearinghouse
  attribute information
    displaying  592
    displaying, cached  580
  create directory replica  539
  listing  556
    preferred
      displaying  584
    specifying  570
clearinghouse catalog command  101
clearinghouse command  98
clearinghouse create command  102
clearinghouse delete  103
clearinghouse disable  104
clearinghouse help command  105
clearinghouse initiate command  106
clearinghouse operations command  107
clearinghouse repair command  108
clearinghouse show command  109
clearinghouse verify command  111
clearinghouses
  creating  534
  deleting  543
  making available  534
clerk
  attribute information, obtaining  595
  stopping  548
clerk cache
  create access to server  540
  displaying contents  551
  displaying server address  582
  removing access to server  531
clerk call confidence level
  displaying  583
  setting  569
clerk or server configuration, DTS  612
clock adjustment
  synchronize  635
  update  637
clock command  112
clock compare command  113
clock help command  114
clock operations command  115
clock set command  116
clock show command  117
clock synchronize command  118
command history and command line recall, dcecp  136
Commands, loginact  656
Commands, pwd_strengthd  663
commands, secadmin  676
compiler, IDL, calling  453
compiling IDL files  458
configuring DCE  2
configuring system as global server  610
convenience variables, dcecp  132
create child command  533
create command  612
create directory command  535
create link command  536
create object command  538
create replica command  539
creating security server, registry database  685
cross linking RACF and DCE users  659, 661
mvsexpt  659
mvsimpt  661

D
daemon, ACL security process, starting  688
database file for endpoint map
query  466
rebuild  469
DCE clock, setting  9
DCE commands
dceconf  2
modify
clock  9
daemon  3
debug  7
DCE daemons
CDS
advertiser  3, 508
clerk  3, 508
DTS
time provider  3
time service  3
level  5
querying  6
RPC  3
Security Service client  3
start options  3
starting and stopping  5
DCE host daemon
daemon
start options  3
DCE level
querying  6
dcecp
abbreviations  129
arguments lists  130
attribute lists  131
attribute options  131
binding information  132
command history and command line recall  136
command processing  128
dcecp (continued)
convenience variables  132
debugging  133
error handling  133
help  134
lists of lists  131
miscellaneous dcecp commands  124
running operating system commands from
dcecp  128
syntax help  130
Tcl notes  127
dcecp command  119
miscellaneous commands
echo  124
texit  125
login  125
logout  126
quit  126
resolve  126
shell  126
dcecp command processing  128
dcecp commands
account  13
account catalog  17
account create  19
account delete  21
account generate  22
account help  23
account modify  24
account operations  26
account show  27
acl  29
acl check command  35
acl delete  36
acl help  37
acl modify  38
acl operations  40
acl permissions  41
acl replace  42
acl show  44
attrlist  46
attrlist add  47
attrlist getvalues  48
attrlist help  49
attrlist list  50
attrlist modify  51
attrlist operations  52
attrlist remove  53
aud  54
aud disable  56
aud enable  57
aud help  58
aud modify  59
aud operations  60
aud rewind  61
aud show  62

722  Command Reference
dcecp commands (continued)

aud stop 63
audevents 64
audevents catalog 65
audevents help 66
audevents operations 67
audevents show 68
audfilter 69
audfilter catalog 71
audfilter create 72
audfilter delete 73
audfilter help 74
audfilter modify 75
audfilter operations 76
audfilter show 77
audtrail 78
audtrail help 79
audtrail operations 80
audtrail show 81
cdscontext 82
cdscontext create 83
cdscontext delete 85
cdscontext dump 86
cdscontext help 87
cdscontext operations 88
cdscontext show 89
cell 91
cell help 92
cell operations 93
cell ping 94
cell show 96
clearinghouse 98
clearinghouse catalog 101
clearinghouse create 102
clearinghouse delete 103
clearinghouse disable 104
clearinghouse help 105
clearinghouse initiate 106
clearinghouse operations 107
clearinghouse repair 108
clearinghouse show 109
clearinghouse verify 111
clock 112
clock compare 113
clock help 114
clock operations 115
clock set 116
clock show 117
clock synchronize 118
dcecp 119
dcecp object 212, 220, 230, 268, 399
decep object 91
directory 139
directory add 142
directory create 143
directory delete 145
dcecp commands (continued)
directory help 147
directory list 148
directory merge 149
directory modify 151
directory operations 153
directory remove 154
directory show 155
directory synchronize 157
dts 158
dts activate 164
dts catalog 165
dts configure 166
dts deactivate 167
dts help 168
dts modify 169
dts operations 170
dts show 171
dts stop 173
dts synchronize 174
echo 124
endpoint 175
endpoint create 178
endpoint delete 180
endpoint help 182
endpoint operations 183
endpoint show 184
endpoint wlb_activate 186
endpoint wlb_ae_show 188
endpoint wlb_off 190
endpoint wlb_on 192
endpoint wlb_show 194
errtext 125
group 91, 196, 399
group add 199
group catalog 200
group create 201
group delete 203
group help 204
group list 205
group modify 206
group operations 208
group remove 209
group rename 210
group show 211
host 212
host catalog 213
host help 214
host operations 215
host ping 216
host show 217
host start 218
host stop 219
hostdata 220
hostdata catalog 222
hostdata create 223
dcecp commands (continued)
hostdata delete 225
hostdata help 226
hostdata modify 227
hostdata operations 228
hostdata show 229
keytab 230
keytab add 232
keytab catalog 234
keytab create 235
keytab delete 237
keytab help 238
keytab list 239
keytab operations 240
keytab remove 241
keytab show 242
link 243
link create 245
link delete 247
link help 248
link modify 249
link operations 250
link show 251
log 252
log help 256
log list 257
log modify 258
log operations 259
log show 260
login 125
logout 126
name 261
name compare 262
name expand 263
name get 264
name help 265
name operations 266
name parse 267
object 268
object create 270
object delete 271
object help 272
object modify 273
object operations 275
object show 276
organization 278
organization add 281
organization catalog 282
organization create 283
organization delete 285
organization help 286
organization list 287
organization modify 288
organization operations 290
organization remove 291
organization rename 293
dcecp commands (continued)
orrganization show 295
principal 297
principal catalog 299
principal create 300
principal delete 302
principal help 303
principal modify 304
principal operations 306
principal rename 307
principal show 308
quit 126
registry 309
registry catalog 315
registry connect 316
registry delete 318
registry designate 319
registry destroy 320
registry disable 321
registry dump 322
registry enable 324
registry help 325
registry modify 326
registry operations 327
registry show 328
registry stop 330
registry synchronize 331
registry verify 332
resolve 126
rpcentry 333
rpcentry create 334
rpcentry delete 335
rpcentry export 336
rpcentry help 338
rpcentry import 339
rpcentry operations 341
rpcentry show 342
rpcentry unexport 344
rpcgroup 346
rpcgroup add 348
rpcgroup create 349
rpcgroup delete 350
rpcgroup help 351
rpcgroup import 352
rpcgroup list 354
rpcgroup operations 356
rpcgroup remove 357
rpcprofile 358
rpcprofile add 360
rpcprofile create 362
rpcprofile delete 363
rpcprofile help 364
rpcprofile import 365
rpcprofile list 367
rpcprofile operations 368
rpcprofile remove 369
Index
dts operations command  170
DTS server  
  creating  612  
  deleting  613  
  disabling  614  
  enabling  619  
DTS server creating  612  
DTS server deleting  613  
DTS server disabling  614  
DTS server enabling  619  
dts show command  171  
dts stop command  173  
dts synchronize command  174  
dtscp  616  
dtscp (DTS control program)  
  attributes  618  
  starting  615  
dtscp commands  
  advertise  610  
  change  611  
  create  612  
  delete  613  
  disable  614  
  dtscp command syntax  616  
  enable  619  
  exit  620  
  help  621  
  quit  622  
  set  623  
  show  627  
  synchronize  635  
  unadvertise  636  
  update  637  
dtscp daemon  
  start options  4  
dtstp daemon  
  start options  4  
dump clerk cache command  551  

debugging  
  dcecp command  124  
editing registry database  664  
elements  
  displaying  494  
  removing  489  
  removing from endpoint map  485  
  removing, select  481  
enable command  619  
enabling disabled time service entity  619  
enabling disabled time service entity enabling disabled time service entity  619  
endpoint command  175  
endpoint create command  178  
endpoint delete command  180  
endpoint help command  182  
endpoint map  
  database file  
    query  466  
    rebuild  469  
  description, adding  443  
  elements  
    displaying  494  
endpoint map (continued)  
  elements (continued)  
    removing  485  
    managing  471  
endpoint operations command  183  
endpoint show command  184  
endpoint wlb_activate command  186  
endpoint wlb_ae_show command  188  
endpoint wlb_off command  190  
endpoint wlb_on command  192  
endpoint wlb_show command  194  
endpoint workload balancing command  175  
entering CDS internal commands  528  
environment variables, RPC  
  RPC_DEFAULT_ENTRY  475  
  RPC_DEFAULT_ENTRY_SYNTAX  475  
epoch number, changing  611  
error handling and debugging, dcecp  133  
errtext dcecp command  125  
ext command  447  
ext command (dtscp)  620  
exting from DTS control program  620  
expired user account  649  
export command  448  
exporting, registry database  690  

F  
format for timestamps  618  

G  
gds daemon  
  start options  4  
Global Directory Agent (GDA)  
  daemons  
    start options  4  
global name  475  
global server  
  adding to cell profile  610  
  removing entry from cell profile  636  
glossary  697  
group add command  199  
group catalog command  200  
group command  196  
group create command  201  
group delete command  203  
group help command  204  
group list command  205  
group members  
  removing  
    all  484  
    specified  487  
  showing  492  
group modify command  206  


echo dcecp command  124  
editing registry database  664  
elements  
  displaying  494  
  removing  489  
  removing from endpoint map  485  
  removing, select  481  
enable command  619  
enabling disabled time service entity  619  
endpoint command  175  
endpoint create command  178  
endpoint delete command  180  
endpoint help command  182  
endpoint map  
  database file  
    query  466  
    rebuild  469  
  description, adding  443  
  elements  
    displaying  494  
endpoint map (continued)  
  elements (continued)  
    removing  485  
    managing  471  
endpoint operations command  183  
endpoint show command  184  
endpoint wlb_activate command  186  
endpoint wlb_ae_show command  188  
endpoint wlb_off command  190  
endpoint wlb_on command  192  
endpoint wlb_show command  194  
endpoint workload balancing command  175  
entering CDS internal commands  528  
environment variables, RPC  
  RPC_DEFAULT_ENTRY  475  
  RPC_DEFAULT_ENTRY_SYNTAX  475  
epoch number, changing  611  
error handling and debugging, dcecp  133  
errtext dcecp command  125  
ext command  447  
ext command (dtscp)  620  
exting from DTS control program  620  
expired user account  649  
export command  448  
exiting, registry database  690  

F  
format for timestamps  618  

G  
gds daemon  
  start options  4  
Global Directory Agent (GDA)  
  daemons  
    start options  4  
global name  475  
global server  
  adding to cell profile  610  
  removing entry from cell profile  636  
glossary  697  
group add command  199  
group catalog command  200  
group command  196  
group create command  201  
group delete command  203  
group help command  204  
group list command  205  
group members  
  removing  
    all  484  
    specified  487  
  showing  492  
group modify command  206
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>group operation command</td>
<td>208</td>
</tr>
<tr>
<td>group remove command</td>
<td>209</td>
</tr>
<tr>
<td>group rename command</td>
<td>210</td>
</tr>
<tr>
<td>group show command</td>
<td>211</td>
</tr>
<tr>
<td>group, naming</td>
<td>478</td>
</tr>
<tr>
<td>help command (cdscp)</td>
<td>552</td>
</tr>
<tr>
<td>help command (dtscp)</td>
<td>621</td>
</tr>
<tr>
<td>help command (rpccp)</td>
<td>451</td>
</tr>
<tr>
<td>help, dcecp</td>
<td>134</td>
</tr>
<tr>
<td>host catalog command</td>
<td>213</td>
</tr>
<tr>
<td>host command</td>
<td>212</td>
</tr>
<tr>
<td>host help command</td>
<td>214</td>
</tr>
<tr>
<td>host operations command</td>
<td>215</td>
</tr>
<tr>
<td>host ping command</td>
<td>216</td>
</tr>
<tr>
<td>host show command</td>
<td>217</td>
</tr>
<tr>
<td>host start command</td>
<td>218</td>
</tr>
<tr>
<td>host stop command</td>
<td>219</td>
</tr>
<tr>
<td>hostdata catalog command</td>
<td>222</td>
</tr>
<tr>
<td>hostdata command</td>
<td>220</td>
</tr>
<tr>
<td>hostdata create command</td>
<td>223</td>
</tr>
<tr>
<td>hostdata delete command</td>
<td>225</td>
</tr>
<tr>
<td>hostdata help command</td>
<td>226</td>
</tr>
<tr>
<td>hostdata modify command</td>
<td>227</td>
</tr>
<tr>
<td>hostdata operations command</td>
<td>228</td>
</tr>
<tr>
<td>hostdata show command</td>
<td>229</td>
</tr>
<tr>
<td>keytab operations command</td>
<td>240</td>
</tr>
<tr>
<td>keytab remove command</td>
<td>241</td>
</tr>
<tr>
<td>keytab show command</td>
<td>242</td>
</tr>
<tr>
<td>kinit command</td>
<td>652</td>
</tr>
<tr>
<td>klist command</td>
<td>654</td>
</tr>
<tr>
<td>ldap_addcell utility</td>
<td>511</td>
</tr>
<tr>
<td>link command</td>
<td>243</td>
</tr>
<tr>
<td>link create command</td>
<td>245</td>
</tr>
<tr>
<td>link delete command</td>
<td>247</td>
</tr>
<tr>
<td>link help command</td>
<td>248</td>
</tr>
<tr>
<td>link modify command</td>
<td>249</td>
</tr>
<tr>
<td>link operations command</td>
<td>250</td>
</tr>
<tr>
<td>link show command</td>
<td>251</td>
</tr>
<tr>
<td>link, soft</td>
<td></td>
</tr>
<tr>
<td>local time</td>
<td></td>
</tr>
<tr>
<td>login</td>
<td></td>
</tr>
<tr>
<td>login account</td>
<td></td>
</tr>
<tr>
<td>login activity</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>552</td>
</tr>
<tr>
<td>help (cdscp)</td>
<td>621</td>
</tr>
<tr>
<td>help (dtscp)</td>
<td>621</td>
</tr>
<tr>
<td>help (rpccp)</td>
<td>451</td>
</tr>
<tr>
<td>host catalog</td>
<td>213</td>
</tr>
<tr>
<td>host</td>
<td>212</td>
</tr>
<tr>
<td>host help</td>
<td>214</td>
</tr>
<tr>
<td>host operations</td>
<td>215</td>
</tr>
<tr>
<td>host ping</td>
<td>216</td>
</tr>
<tr>
<td>host show</td>
<td>217</td>
</tr>
<tr>
<td>host start</td>
<td>218</td>
</tr>
<tr>
<td>host stop</td>
<td>219</td>
</tr>
<tr>
<td>hostdata catalog</td>
<td>222</td>
</tr>
<tr>
<td>hostdata</td>
<td>220</td>
</tr>
<tr>
<td>hostdata create</td>
<td>223</td>
</tr>
<tr>
<td>hostdata delete</td>
<td>225</td>
</tr>
<tr>
<td>hostdata help</td>
<td>226</td>
</tr>
<tr>
<td>hostdata modify</td>
<td>227</td>
</tr>
<tr>
<td>hostdata operations</td>
<td>228</td>
</tr>
<tr>
<td>hostdata show</td>
<td>229</td>
</tr>
<tr>
<td>keytab</td>
<td>240</td>
</tr>
<tr>
<td>keytab remove</td>
<td>241</td>
</tr>
<tr>
<td>keytab show</td>
<td>242</td>
</tr>
<tr>
<td>kinit</td>
<td>652</td>
</tr>
<tr>
<td>klist</td>
<td>654</td>
</tr>
<tr>
<td>ldap_addcell</td>
<td>511</td>
</tr>
<tr>
<td>link</td>
<td>243</td>
</tr>
<tr>
<td>link create</td>
<td>245</td>
</tr>
<tr>
<td>link delete</td>
<td>247</td>
</tr>
<tr>
<td>link help</td>
<td>248</td>
</tr>
<tr>
<td>link modify</td>
<td>249</td>
</tr>
<tr>
<td>link operations</td>
<td>250</td>
</tr>
<tr>
<td>link show</td>
<td>251</td>
</tr>
<tr>
<td>link, soft</td>
<td></td>
</tr>
<tr>
<td>local time</td>
<td></td>
</tr>
<tr>
<td>login</td>
<td></td>
</tr>
<tr>
<td>login account</td>
<td></td>
</tr>
<tr>
<td>login activity</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>calling IDL compiler</td>
<td>453</td>
</tr>
<tr>
<td>compiling IDL files to output stub format</td>
<td>458</td>
</tr>
<tr>
<td>options</td>
<td>453</td>
</tr>
<tr>
<td>under the shell</td>
<td>461</td>
</tr>
<tr>
<td>under TSO/E</td>
<td>461</td>
</tr>
<tr>
<td>import command</td>
<td>463</td>
</tr>
<tr>
<td>importing</td>
<td>663</td>
</tr>
<tr>
<td>importing, registry database</td>
<td>691</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>kdestroy command</td>
<td>650</td>
</tr>
<tr>
<td>Kerberos tickets</td>
<td></td>
</tr>
<tr>
<td>listing</td>
<td>654</td>
</tr>
<tr>
<td>obtaining</td>
<td>652</td>
</tr>
<tr>
<td>key, master</td>
<td>676</td>
</tr>
<tr>
<td>keytab add command</td>
<td>232</td>
</tr>
<tr>
<td>keytab catalog command</td>
<td>234</td>
</tr>
<tr>
<td>keytab command</td>
<td>230</td>
</tr>
<tr>
<td>keytab create command</td>
<td>235</td>
</tr>
<tr>
<td>keytab delete command</td>
<td>237</td>
</tr>
<tr>
<td>keytab help command</td>
<td>238</td>
</tr>
<tr>
<td>keytab list command</td>
<td>239</td>
</tr>
<tr>
<td>keytab operations</td>
<td>240</td>
</tr>
<tr>
<td>keytab remove</td>
<td>241</td>
</tr>
<tr>
<td>keytab show</td>
<td>242</td>
</tr>
<tr>
<td>kinit</td>
<td>652</td>
</tr>
<tr>
<td>klist</td>
<td>654</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>calling IDL compiler</td>
<td>453</td>
</tr>
<tr>
<td>compiling IDL files to output stub format</td>
<td>458</td>
</tr>
<tr>
<td>options</td>
<td>453</td>
</tr>
<tr>
<td>under the shell</td>
<td>461</td>
</tr>
<tr>
<td>under TSO/E</td>
<td>461</td>
</tr>
<tr>
<td>import command</td>
<td>463</td>
</tr>
<tr>
<td>importing</td>
<td>663</td>
</tr>
<tr>
<td>importing, registry database</td>
<td>691</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>kdestroy command</td>
<td>650</td>
</tr>
<tr>
<td>Kerberos tickets</td>
<td></td>
</tr>
<tr>
<td>listing</td>
<td>654</td>
</tr>
<tr>
<td>obtaining</td>
<td>652</td>
</tr>
<tr>
<td>key, master</td>
<td>676</td>
</tr>
<tr>
<td>keytab add command</td>
<td>232</td>
</tr>
<tr>
<td>keytab catalog command</td>
<td>234</td>
</tr>
<tr>
<td>keytab command</td>
<td>230</td>
</tr>
<tr>
<td>keytab create command</td>
<td>235</td>
</tr>
<tr>
<td>keytab delete command</td>
<td>237</td>
</tr>
<tr>
<td>keytab help command</td>
<td>238</td>
</tr>
<tr>
<td>keytab list command</td>
<td>239</td>
</tr>
<tr>
<td>keytab operations</td>
<td>240</td>
</tr>
<tr>
<td>keytab remove</td>
<td>241</td>
</tr>
<tr>
<td>keytab show</td>
<td>242</td>
</tr>
<tr>
<td>kinit</td>
<td>652</td>
</tr>
<tr>
<td>klist</td>
<td>654</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>calling IDL compiler</td>
<td>453</td>
</tr>
<tr>
<td>compiling IDL files to output stub format</td>
<td>458</td>
</tr>
<tr>
<td>options</td>
<td>453</td>
</tr>
<tr>
<td>under the shell</td>
<td>461</td>
</tr>
<tr>
<td>under TSO/E</td>
<td>461</td>
</tr>
<tr>
<td>import command</td>
<td>463</td>
</tr>
<tr>
<td>importing</td>
<td>663</td>
</tr>
<tr>
<td>importing, registry database</td>
<td>691</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>kdestroy command</td>
<td>650</td>
</tr>
<tr>
<td>Kerberos tickets</td>
<td></td>
</tr>
<tr>
<td>listing</td>
<td>654</td>
</tr>
<tr>
<td>obtaining</td>
<td>652</td>
</tr>
<tr>
<td>key, master</td>
<td>676</td>
</tr>
<tr>
<td>keytab add command</td>
<td>232</td>
</tr>
<tr>
<td>keytab catalog command</td>
<td>234</td>
</tr>
<tr>
<td>keytab command</td>
<td>230</td>
</tr>
<tr>
<td>keytab create command</td>
<td>235</td>
</tr>
<tr>
<td>keytab delete command</td>
<td>237</td>
</tr>
<tr>
<td>keytab help command</td>
<td>238</td>
</tr>
<tr>
<td>keytab list command</td>
<td>239</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>keytab</td>
<td>240</td>
</tr>
<tr>
<td>keytab remove</td>
<td>241</td>
</tr>
<tr>
<td>keytab show</td>
<td>242</td>
</tr>
<tr>
<td>kinit</td>
<td>652</td>
</tr>
<tr>
<td>klist</td>
<td>654</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>calling IDL compiler</td>
<td>453</td>
</tr>
<tr>
<td>compiling IDL files to output stub format</td>
<td>458</td>
</tr>
<tr>
<td>options</td>
<td>453</td>
</tr>
<tr>
<td>under the shell</td>
<td>461</td>
</tr>
<tr>
<td>under TSO/E</td>
<td>461</td>
</tr>
<tr>
<td>import command</td>
<td>463</td>
</tr>
<tr>
<td>importing</td>
<td>663</td>
</tr>
<tr>
<td>importing, registry database</td>
<td>691</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>kdestroy command</td>
<td>650</td>
</tr>
<tr>
<td>Kerberos tickets</td>
<td></td>
</tr>
<tr>
<td>listing</td>
<td>654</td>
</tr>
<tr>
<td>obtaining</td>
<td>652</td>
</tr>
<tr>
<td>key, master</td>
<td>676</td>
</tr>
<tr>
<td>keytab add command</td>
<td>232</td>
</tr>
<tr>
<td>keytab catalog command</td>
<td>234</td>
</tr>
<tr>
<td>keytab command</td>
<td>230</td>
</tr>
<tr>
<td>keytab create command</td>
<td>235</td>
</tr>
<tr>
<td>keytab delete command</td>
<td>237</td>
</tr>
<tr>
<td>keytab help command</td>
<td>238</td>
</tr>
<tr>
<td>keytab list command</td>
<td>239</td>
</tr>
</tbody>
</table>

Index 727
logout dcecp command 126

M
macro, data set execution in CDSCP 550
map database, adding endpoint description 443
master replica 676
miscellaneous dcecp commands
echo 124
mkdcereregister utility 513
modify
clock 9
daemon 3
debug 7
modifying time service entity profile 623
mvsexpt
cross linking with 659
mvsexpt command 659
mvsimpt
cross linking with 661
mvsimpt command 661

N
name command 261
name compare command 262
name entry, removing from database 483
name expand command 263
name get command 264
name help command 265
name operations command 266
name parse command 267
name service
tentries 475
entry name guidelines 476
entry name structure 475
interface 475
Name Service Database
tentry
adding 441
removing 483
removing group and group members 484
showing 490
specifying for removal from group 487
tmembers
adding to entry 445
showing 492
tprofile element
adding 438
removing 481, 489
showing 497
tnaming
tgroup 478
tprofile 478
tserver entry 477

O
object command 268
object create command 270
object delete command 271
object entry
adding attribute value 517
changing attribute value 578
creating 538
deleting 546
displaying attribute information 602
listing 562
removing attribute value 567
object help command 272
object modify command 273
object operations command 275
object show command 276
organization add command 281
organization catalog command 282
organization command 278
organization create command 283
organization delete command 285
organization help command 286
organization list command 287
organization modify command 288
organization operations command 290
organization remove command 291
organization rename command 293
organization show command 295

P
password
creating 673
deleting 673
displaying 673
password storing 693
passwords, encrypting 676
ping command 465
pointers, child
creating 533
deleting 542
displaying attribute information 590
listing 555
principal catalog command 299
principal command 297
principal create command 300
principal delete command 302
principal help command 303
principal modify command 304
principal operations command 306
principal rename command 307
principal show command 308
profile element
  adding 438
  description of fields 439
  removing
    all 489
    select 481
    showing 497
profile, naming 478
pwd_strengthd command 663

Q
query epdb 466
query epdb command 466
quit 468
quit command 468
quit command (dtscp) 622
quit dcecp command 126

R
rebuild epdb 469
rebuild epdb command 469
registry catalog command 315
registry command 309
registry connect command 316
registry database, exporting 690
registry database, importing 691
registry database, viewing and editing 664
registry delete command 318
registry designate command 319
registry destroy command 320
registry disable command 321
registry dump command 322
registry enable command 324
registry entry
  add a new name 665
  changing a principal, group, or organization 666
  creating a principal, group, or organization 667
  deleting 667
  editing membership list 666
  viewing 665
registry help command 325
registry modify command 326
registry operations command 327
registry show command 328
registry stop command 330
registry synchronize command 331
registry verify command 332
registry, changing or displaying
  authentication policies 675
  defaults 675
  properties 674
  site 674
  standard policies 674
  type of information 674
Remote Procedure Call (RPC)
  control program (rpccp)
    continuation character 472
    entry name structure 475
    environment variables 475
    exiting 447
    getting help 451
    name construction 476
    name service entries 475
    name service interface 475
    options 472
    privilege required 478
    quitting 468
    scope of commands 474
    starting 471
  introduction 437
remove directory command 564
remove element command 481
remove entry command 483
remove group command 484
remove link command 566
remove mapping command 485
remove member command 487
remove object command 567
remove profile command 489
removing principal's login ID 650
replica 676
  attribute information, obtaining 604
  creating 539
  deleting 547
  making consistent 575
  reconstructing 573
remove directory command 564
remove element command 481
remove entry command 483
remove group command 484
remove link command 566
remove mapping command 485
remove member command 487
remove object command 567
remove profile command 489
removing principal's login ID 650
replica 676
  attribute information, obtaining 604
  creating 539
  deleting 547
  making consistent 575
  reconstructing 573
 resolve dcecp command 126
rgy_edit
  See rgyedit
rgyedit
  displaying usage information 675
  establishing new identity 675
  exiting 675
  limiting information displayed 675
  starting 665
rgyedit command 664
rgyedit subcommands
  account
    add 668
    cell 670
    change 670
    delete 670
    view 668
  key management
    ktadd 673
    ktdelte 673
    ktlst 673
miscellaneous
  authpolicy 675
  defaults 675
  domain 674

Index 729
rgyedit subcommands (continued)
  miscellaneous (continued)
    exit 675
    help 675
    login 675
    policy 674
    properties 674
    quit 675
    scope 675
    site 674
    principals, groups, and organizations
      add 665
      adopt 667
      change 666
      delete 667
      member 666
      view 665
  routing control file 252, 254, 255
  routing for serviceability messages 252
  routing serviceability debug messages 255
  RPC_DEFAULT_ENTRY 475
  RPC_DEFAULT_ENTRY_SYNTAX 475
  rpccp (RPC control program)
    continuation character 472
    entry name structure 475
    environment variables 475
    exiting 447
    getting help 451
    name construction 476
    name service entries 475
    name service interface 475
    options 472
    privilege required 478
    quitting 468
    scope of commands 474
    starting 471
  rpccp command 470
  rpccp commands
    add element 438
    add entry 441
    add mapping 443
    add member 445
    exit 447
    export 448
    help 451
    IDL 453
    import 463
    ping 465
    query epdb 466
    quit 468
    rebuild epdb 469
    remove element 481
    remove entry 483
    remove group 484
    remove mapping 485
    remove member 487
  rpccp commands (continued)
    remove profile 489
    rpccp 470
    show entry 490
    show group 492
    show mapping 494
    show profile 497
    show server 500
    unexport 502
    uuidgen 504
  rpcentry command 333
  rpcentry create command 334
  rpcentry delete command 335
  rpcentry export command 336
  rpcentry help command 338
  rpcentry import command 339
  rpcentry operations command 341
  rpcentry show command 342
  rpcentry unexport command 344
  rpcgroup add command 348
  rpcgroup command 346
  rpcgroup create command 349
  rpcgroup delete command 350
  rpcgroup help command 351
  rpcgroup import command 352
  rpcgroup list command 354
  rpcgroup operations command 356
  rpcgroup remove command 357
  rpcprofile add command 360
  rpcprofile command 358
  rpcprofile create command 362
  rpcprofile delete command 363
  rpcprofile help command 364
  rpcprofile import command 365
  rpcprofile list command 367
  rpcprofile operations command 368
  rpcprofile remove command 369
  rpcprofile show command 371
  running operating system commands from dcecp 128

S
  schema command 424
  sec_create_db command 685
  sec_eimort_db command 691
  sec_export_db command 690
  sec_import_db command 691
  secadmin command 676
  secd command 688
  secd daemon
    start options 3
  Security Server 676
  Security Service
    daemon
      start options 3
Security Service (continued)

introduction 639

Security Service commands

supported
acl_edit 641
acledit 641
dce_login 648
dcelogin 648
kdestroy 650
kinit 652
klist 654
log_inact 656
loginact 656
mvsexpt 659
mvsimpt 661
pwd_strengthd 663
rgy_edit 664
rgyedit 664
sec_admin 676
sec_create_db 685
sec_export_db 690
sec_import_db 691
secadmin 676
secd 688
storepw 693

unsupported
chpass 639
passwd_export 639
passwd_import 639
passwd_override 639
sec_salvage_db 639

secval activate command 375
secval command 374
secval deactivate command 376
secval help command 377
secval operations command 378
secval ping command 379
secval status command 380

server
displaying address in clerk's cache 582
enable access through clerk's cache 540
removing access from local clerk's cache 531

server binding information
exporting 448
importing 463
showing 500
unexporting 502
server catalog command 385
server command 381
server create command 386
server delete command 388
server disable command 389
server enable command 390
server entity 607
server entry, naming 477
server help command 391
server modify command 392
server operations command 394
server ping command 395
server show command 396
server start command 397
server stop command 398
serviceability 252
serviceability debug message levels, setting 254
serviceability messages, routing 252
set cdscp confidence command 569
set cdscp preferred clearinghouse command 570
set command 623
set directory command 571
set directory to new epoch command 573
set directory to skulk command 575
set link command 576
set object command 578
shell dcecp command 126
show cached clearinghouse command 580
show cached server command 582
show cdscp confidence command 583
show cdscp preferred clearinghouse command 584
show cdscp syntax command 585
show cell command 588
show child command 590
show clearinghouse command 592
show clerk command 595
show command 627
show directory command 597
show entry command 490
show group command 492
show link command 600
show mapping command 494
show object command 602
show profile command 497
show replica command 604
show server command 500
skulk, starting 575
slave replica 676
soft link
attribute information, obtaining 600
changing attribute value 576
creating 536
creating list with name match search 560
deleting 545
making permanent 566
starting DCE 2
starting security server process daemon 688
storepw command 693
svcroute 252, 254
synchronize command 635
synchronizing time 635
syntax help, dcecp 130

Index 731
syntax, cdscp 585

T
Tcl notes 127
ticket-granting ticket (Kerberos)
  listing 654
  obtaining 652
time service
  clerk or server
    creating 612
    deleting 613
    disabling 614
    enabling 619
  control program, quitting 622
  local profile, setting 623
time synchronization, setting to midpoint 635
timestamp format 618

U
unadvertise command 636
  unadvertise 636
unexport command 502
Universal Unique Identifier (UUID)
  object
    binding information, obtaining from server 463
    binding information, sending to server 448
  server entry
    displaying object UUID 500
    removing object UUID 502
update command 637
update registry, master database, creating 685
user account expiration 649
user command 399
user create command 403
user delete command 406
user help command 407
user operations command 408
user show command 409
utc add command 412
utc command 411
utc compare command 413
utc convert command 414
utc help command 415
utc multiply command 416
utc operations command 417
utc subtract command 418
uuid command 419
uuid compare command 420
uuid create command 421
uuid help command 422
uuid operations command 423
uuidgen command 504

V
validating principal’s login ID 648

W
workload balancing 186
workload balancing, endpoint command 175
workload management (WLM) 175

X
xattrschema catalog command 428
xattrschema command 424
xattrschema create command 429
xattrschema delete command 430
xattrschema help command 431
xattrschema modify command 432
xattrschema operations command 434
xattrschema rename command 435
xattrschema show command 436
Readers' Comments

z/OS
DCE
Command Reference
Publication No. SC24-5909-00

You may use this form to report errors, to suggest improvements, or to express your opinion on the appearance, organization, or completeness of this book.

Date: ________________________________

__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note

Report system problems to your IBM representative or the IBM branch office serving you. U.S. customers can order publications by calling the IBM Software Manufacturing Solutions at 1-800-879-2755.

In addition to using this postage-paid form, you may send your comments by:

FAX 1-607-752-2327
IBMLink GDLVME(PUBRCF)

Internet pubrcf@vnet.ibm.com

Would you like a reply?  ___YES  ___NO  If yes, please tell us the type of response you prefer.

___ Electronic address: ________________________________

___ FAX number: ________________________________

___ Mail: (Please fill in your name and address below.)

Name

Address

Company or Organization

Phone No.